UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Sciences et Ingénierie pour l'Information
FACULTÉ des SCIENCES et TECHNIQUES
Département de Mathématiques et Informatique
Laboratoire XLIM (UMR 6172)

Thèse N° 58 - 2011

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline : Mathématiques et ses applications

présentée et soutenue publiquement par

Carole EL BACHA

le 25 novembre 2011 à 14h

Méthodes Algébriques pour la Résolution d'Équations Différentielles Matricielles d'Ordre Arbitraire

dirigée par Moulay A. BARKATOU et codirigée par Thomas CLUZEAU

devant le jury composé de

Rapporteurs :

Mark VAN HOEIJ	Professeur, Université d'État de Floride
George LABAHN	Professeur, Université de Waterloo
Viktor LEVANDOVSKYY	Assistant-Professeur, Université d'Aix-La-Chapelle

Examinateurs :

Sergei ABRAMOV	Professeur, Université de Moscou
Moulay A. BARKATOU	Professeur, Université de Limoges
Alin BOSTAN	Chargé de Recherche, INRIA Rocquencourt
François BOULIER	Professeur, Université Lille I
José-Maria CANO TORRES	Professeur, Université de Valladolid
Thomas CLUZEAU	Maître de Conférences, Université de Limoges

À mes parents,
À ma sœur et mon frère,
À mes neveux.

Remerciements

Cette thèse ne m'a pas juste apporté le titre de docteur mais elle m'a aussi permis de m'enrichir personnellement, socialement, intellectuellement et scientifiquement. Je profite de ces quelques lignes pour remercier les personnes qui ont contribué à la réussite de ce travail.

J'ai eu de la chance d'avoir un directeur de thèse et un co-encadrant formidables. Je parle de Moulay Barkatou et Thomas Cluzeau sans qui ce travail n'aurait jamais vu le jour. Je tiens à les remercier très chaleureusement pour leur assistance, leur disponibilité, leur patience et leurs précieux conseils. Je voudrais aussi les remercier pour le temps qu'ils m'ont consacré tout au long de ces années et la pleine confiance qu'ils m'ont accordée tout au long de ce travail. Leur gentillesse, leur soutien et leur encouragement étaient mon remède pour rebondir de mes faiblesses et aller plus loin. Leur perfectionnisme m'a poussée à donner le meilleur de moi-même. Je leur suis très reconnaissante de m'avoir mise en contact avec des gens en dehors du laboratoire XLIM, ce qui m'a permis de mieux nourrir mes réflexions et enrichir mes connaissances dans le monde du Calcul Formel. Enfin, merci pour tous les bons moments passés ensemble lors de nos voyages en mission, je ne les oublierai jamais.

Je suis très reconnaissante envers Mark van Hoeij, George Labahn et Viktor Levandovskyy d'avoir accepté de rapporter sur ma thèse. Je les remercie pour le temps qu'ils ont consacré pour bien lire ce manuscrit, pour leurs remarques constructives et pour leurs jugements pertinents. Je remercie plus particulièrement George avec qui j'ai eu une collaboration fructueuse présentée dans cette thèse. Merci aussi à Viktor pour sa sympathie (et surtout pour la boîte de chocolat).

Je remercie également Sergei Abramov, Alin Bostan, François Boulier et José-Maria Cano Torres d'avoir accepté d'examiner cette thèse et fait le voyage pour participer à mon jury.

Ma reconnaissance va également à Eckhard Pflügel avec qui j'ai co-signé deux articles. Je le remercie pour l'intérêt qu'il a su montrer à cette thèse.

Mes plus sincères remerciements vont aussi aux membres de l'équipe Algorithms de l'INRIA Paris-Rocquencourt et ceux de l'équipe Calcul Formel du Laboratoire d'Informatique Fondamentale de Lille de m'avoir conviée à présenter mes travaux de thèse dans leurs établissements. Je les remercie pour leurs nombreux conseils et les discussions intéressantes que nous avons pu avoir.

Je n'oublie pas Clemens Raab, Flavia Stan et Morgan Barbier pour leurs encouragements avant la soutenance et leurs félicitations après.

J'exprime tous mes remerciements aux personnels du Département Mathématiques Informatique (DMI) de l'Université de Limoges de m'avoir accueillie chaleureusement parmi eux. Merci tout particulièrement à mon tuteur pédagogique Alain Salinier de m'avoir guidée dans mes premiers pas dans le monde académique et à Jacques-Arthur Weil avec qui j'ai eu des échanges passionnés. Merci également à Odile Duval, Annie Nicolas, Nadia Rossi, Patricia Vareille et Yolande Vicelli pour leur disponibilité pour répondre à tous nos besoins. Une pensée aux doctorants qui sont passés par Limoges et qui sont devenus docteurs, en particulier Julien Angeli, Aurore Bernard, Christophe Chabot, Daouda Diatta, Sandrine Jean et Benjamin

Pousse. Également aux doctorants Slim Bettaieb, Abdelkarim Chakhar, Ba-Khiet Le, Yves Jonathan Ndje, Hoang Anh Nguyen, Esteban Segura Ugalde, Zhenzhong Song et Gaël Thomas.

Une pensée très chère va à Elsa Bousquet pour son amitié, à Ainhoa Aparicio Monforte pour la belle surprise inattendue le jour de ma soutenance, à Amel Gheffar pour son sourire permanent et sa bonté et à Delphine Poingt pour son amitié depuis le master. J'adresse aussi toute ma gratitude à Marie-Jeanne Nal pour sa gentillesse.

Je tiens également à remercier mes compatriotes qui ont rendu mon séjour à Limoges plus agréable : Hadia Rammal pour son amitié et les longues journées de shopping, et Michel El Khoury et Georges Zakka El Nachef pour tous les bons moments passés ensemble entre autres à la pause déjeuner de midi.

Je remercie aussi et très chaleureusement Hassan Saoud pour ses valeurs humaines, pour la force qu'il me transmettait dans les moments les plus difficiles, pour son soutien et ses encouragements sans faille, pour son aide, pour son écoute et pour pleine d'autres choses.

Enfin, le plus grand merci va à toute ma famille, plus particulièrement, à mes parents Antoine et Kamale qui ont cru en mes capacités et m'ont laissée quitter le nid familial pour réaliser mon rêve de devenir docteur. Merci pour leur prières qui m'ont aidée à aller jusqu'au bout. Difficile en quelques mots de leur exprimer toute ma reconnaissance et tout mon amour. Merci également à ma sœur Carla et sa famille, Imad, Christophe, Mickael et Marc, et à mon frère Hany et sa famile, Jamila, Anthony et Kevin. Je n'oublie pas mes oncles et mes tantes qui me transmettaient leur soutien par Skype et sms.

Carole El Bacha

Notation

$\mathbb{N}$	The set of nonnegative integers
$\mathbb{N}^*$	The set of positive integers
$\mathbb{Z}$	The ring of integers
$\mathbb{Q}$	The field of rational numbers
$\mathbb{C}$	The field of complex numbers
$\overline{\mathbb{K}}$	The algebraic closure of a field $\mathbb{K}$
$\mathbb{A}[\lambda]$	The ring of polynomials in λ over a ring $\mathbb{A}$
$\mathbb{A}[[x]]$	The ring of formal power series in x over a ring $\mathbb{A}$
$\mathbb{K}(\lambda)$	The field of rational functions in λ over a field $\mathbb{K}$
$\mathbb{K}((x))$	The field of formal Laurent series in x over a field $\mathbb{K}$
∂	The standard derivation $\frac{d}{dx}$
ϑ	The Euler derivation $x\,\frac{d}{dx}$
ϑ_k with $k \in \mathbb{N}$	The derivation $x^k\,\vartheta = x^{k+1}\,\partial$
$\mathbb{A}[D]$ with D a derivation	The ring of differential operators $\sum_{i=0}^{n} a_i\,D^i$ with $n \in \mathbb{N}$ and the a_i's in a ring $\mathbb{A}$
$\mathbb{A}^{m\times n}$	The additive group of $m \times n$ matrices with entries in a ring $\mathbb{A}$
$\mathbb{A}^{n\times n}$	The ring of $n \times n$ matrices with entries in a ring $\mathbb{A}$
$\mathrm{GL}_n(\mathbb{A})$	The general linear group of degree n over a ring $\mathbb{A}$
$\mathbb{A}^n$, resp. $\mathbb{A}^{1\times n}$	The additive group of n-dimensional (column) vectors, resp. row vectors, with components in a ring $\mathbb{A}$
$\mathbb{K}^{m\times n}$	The $\mathbb{K}$-vector space of $m \times n$ matrices with entries in a field $\mathbb{K}$
$\mathbb{K}^{n\times n}$	The $\mathbb{K}$-algebra of $n \times n$ matrices with entries in a field $\mathbb{K}$
$\mathrm{GL}_n(\mathbb{K})$	The general linear group of degree n over a field $\mathbb{K}$
$\mathbb{K}^n$, resp. $\mathbb{K}^{1\times n}$	The $\mathbb{K}$-vector space of n-dimensional (column) vectors, resp. row vectors, with components in a field $\mathbb{K}$
$f'(x)$	The first derivative of a function $f(x)$ w.r.t. to x
$f^{(k)}(x)$	The kth derivative of a function $f(x)$ w.r.t. to x
$\sigma(L)$	The spectrum of a square matrix polynomial $L(\lambda)$
$m_a(\lambda_0)$	The algebraic multiplicity of an eigenvalue λ_0
$m_g(\lambda_0)$	The geometric multiplicity of an eigenvalue λ_0
ω	The exponent for the complexity of matrix multiplication [100]
$\deg(p)$	The degree of a polynomial p
d_{λ_0}	The degree of the extension $\mathbb{K}(\lambda_0)$ over $\mathbb{K}$, where $\lambda_0 \in \overline{\mathbb{K}}$
$\dim(E)$	The dimension of a vector space E

0_n	The square zero matrix of size n
I_n	The identity matrix of size n
A^T	The transpose of a matrix/vector A
A^{-1}	The inverse of an invertible square matrix A
$\mathrm{diag}\,(A_1,\, A_2,\, \ldots,\, A_n)$	The (block) diagonal matrix $\begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_n \end{pmatrix}$
$\mathrm{rank}(A)$	The rank of a matrix A
$\ker(A)$	The right nullspace of a matrix A
$\det(A)$	The determinant of a square matrix A
$\mathrm{adj}(A)$	The adjoint of a matrix A
$\mathrm{coldim}(A)$	The number of columns of a matrix A
$A(i,j)$	The (i,j)th entry of a matrix A
$A(i,.)$	The ith row of a matrix A
$A(.,j)$	The jth column of a matrix A
$A_{i,*}$	The ith block row of a block matrix A
$A_{*,j}$	The jth block column of a block matrix A
$\Re(z)$	The real part of a complex number z
$v(f)$ with $f \in \overline{\mathbb{K}((x))}$	The x-adic valuation of f
$\ell c(f)$ with $f \in \overline{\mathbb{K}((x))}$	The coefficient of $x^{v(f)}$ in f
$v(M)$ with $M \in \overline{\mathbb{K}((x))}^{m\times n}$	The x-adic valuation of M defined by $v(M) = \min\{v(M(i,j));\, 1 \le i \le m, 1 \le j \le n\}$
$\ell c(M)$ with $M \in \overline{\mathbb{K}((x))}^{m\times n}$	The coefficient matrix of $x^{v(M)}$ in M
$M_{\mid\nu}$ with $M \in \mathbb{K}[[x]]^{m\times n}$ and $\nu \in \mathbb{N}$	Matrix M truncated at order ν, *i.e.*, if $M = \sum_{i=0}^{\infty} M_i\, x^i$, then $M_{\mid\nu} = \sum_{i=0}^{\nu} M_i\, x^i$
$\lvert\alpha\rvert$ with $\alpha = (\alpha_1, \ldots, \alpha_n) \in (\mathbb{N} \cup \{\pm\infty\})^n$ and $\alpha \notin \{\pm\infty\}^n$	$\lvert\alpha\rvert = \sum_{1\le i\le n \,\mathrm{s.t.}\, \alpha_i \neq \pm\infty} \alpha_i$
$\delta_{i,j}$	The Kronecker delta defined by $\delta_{i,j} = \begin{cases} 1 & \text{if} \quad i = j, \\ 0 & \text{if} \quad i \neq j. \end{cases}$

Contents

Introduction

Nowadays, the theory of differential equations has an essential place in mathematics and contributes to the development of various scientific fields.

Linear ordinary differential equations (ODEs) have been intensively studied in the past years from both theoretical and algorithmic points of view, and considerable progress has been made on this subject, see [37, 8, 59, 94, 10, 26, 36]. However, the mathematical modeling of many problems in chemistry, physics, mechanics and control theory exceeds scalar linear differential equations and gives rise to systems of linear differential equations of arbitrary order, see [34, 71, 73, 81, 82, 89] and references therein. In this thesis, we are interested in the algorithmic treatment of such systems.

We consider a matrix differential equation of order $\ell \geq 1$ of the form

$$\mathcal{A}_\ell(x)\, y^{(\ell)}(x) + \mathcal{A}_{\ell-1}(x)\, y^{(\ell-1)}(x) + \cdots + \mathcal{A}_0(x)\, y(x) = f(x), \tag{1}$$

where x is a complex variable, the $\mathcal{A}_i$'s are $m \times n$ matrices of analytic functions, f is an m-dimensional vector of analytic functions, y is an unknown n-dimensional vector and $y^{(i)}(x) = \frac{d^i y}{dx^i}(x)$.

When $m = n$ and the matrix $\mathcal{A}_\ell(x)$ is invertible, *i.e.*, $\det(\mathcal{A}_\ell(x)) \neq 0$, Equation (1) is called an *explicit differential system* or simply a *system of linear ordinary differential equations*.

Systems of linear ODEs of first-order, generally given in the form

$$\frac{dy}{dx}(x) = A(x)\, y(x) + b(x)$$

with $A(x)$, respectively $b(x)$, an $n \times n$ matrix, respectively an n-dimensional vector, of meromorphic functions, have been widely investigated and several efficient symbolic algorithms have been developed for solving the local problems (around a singularity, *i.e.*, a pole of $A(x)$):

- determination of the nature of singularities (regular or irregular singularities: Moser's algorithm [77, 54, 12, 25]),
- construction of formal solutions (construction of a fundamental solution matrix, regular and irregular solutions [37, 101, 56, 13, 21, 85]),
- computation of certain formal invariants (Katz's invariant [57, 13], Newton's polygon [85], Malgrange's invariant and Gérard & Levelt's invariant [55]).

There also exist many algorithms treating global problems such as:

- construction of global solutions (rational solutions [14], exponential solutions [83]),
- decomposition and factorization (computing a companion block diagonal form [11], computing a block triangular form [22], using the eigenring method [15]), etc.

Nevertheless, algorithms handling directly higher-order explicit differential systems have been less elaborated. The classical approach to deal with such systems consists in converting them

into the first-order system

$$\frac{d}{dx}\begin{pmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(\ell-2)}(x) \\ y^{(\ell-1)}(x) \end{pmatrix} = \begin{pmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ \widetilde{\mathcal{A}}_0(x) & \widetilde{\mathcal{A}}_1(x) & \widetilde{\mathcal{A}}_2(x) & \dots & \widetilde{\mathcal{A}}_{\ell-1}(x) \end{pmatrix} \begin{pmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(\ell-2)}(x) \\ y^{(\ell-1)}(x) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ f(x) \end{pmatrix}, \quad (2)$$

where for $i = 0, \dots, \ell$, $\widetilde{\mathcal{A}}_i(x) = -\mathcal{A}_\ell^{-1}(x)\,\mathcal{A}_i(x)$, then applying the results and the algorithms developed for first-order systems (see for example [29, 66]). But the conversion of a higher-order system into a first-order one has the computational drawback of increasing the size of the problem; for a differential system of size n and order ℓ, the resulting first-order system is of size $n\,\ell$. Another approach dealing with higher-order systems of linear ODEs has been elaborated by Abramov, Bronstein & Khmelnov (see [1, 2, 4, 5]). This time, systems of the form (1) are transformed into matrix recurrence equations. We find also in literature some direct works as for example in [64, 79, 80] but these latter treat only particular case of (1). Thus, the need to treat directly higher-order systems of linear ODEs is the major motivation of this thesis.

When $m \neq n$ or the matrix $\mathcal{A}_\ell(x)$ is singular, systems of the form (1) are known as *linear differential-algebraic equations (DAEs)*. As the name indicates, a system of DAEs is a system composed of ordinary differential equations coupled with purely algebraic equations and hence DAEs differ from systems of ODEs in many aspects. Linear DAEs have been intensively studied numerically: see [46, 67, 73, 88] and references therein. DAEs of first-order are best classified using various concepts of index [67, 89, 90]. The most known is the *differential index* [46] which measures the distance from the so-called *underlying ordinary differential equation*. The latter is a system of ordinary differential equations of first-order computed by differentiating the DAE successively and then using only algebraic manipulations to express y' as function of y and x. In Computer Algebra, a standard technique to treat linear DAEs is to reduce them into systems of "simpler" forms: the Popov normal form [40] (this form is useful for rewriting high order terms with respect to low order terms hence transforming higher-order DAEs into first-order DAEs), the Hermite normal form [49] (an upper triangular matrix whose order is in general larger than the one of the input DAE), the Jacobson normal form [38, 70, 74] (a diagonal matrix which reduces the system into a scalar linear differential equation). An implementation of the algorithms computing these normal forms is provided in the Computer Algebra system Singular [51]. In this thesis, in order to apply the classical theory of ODEs, we are interested in the algorithms that decouple a given linear DAE into a purely differential system and a purely algebraic one.

Content of the Thesis

In this thesis, we are mainly interested in the local analysis systems of the form (1) at a point $x_0 \in \mathbb{C}$ which can be supposed, without any loss of generality, to be the origin, *i.e.*, $x_0 = 0$. Therefore, we will assume that the entries of the coefficient matrices $\mathcal{A}_i(x)$ and of the right-hand side vector $f(x)$ are power series in x.

The thesis is split into two essential parts. In the first part, which is composed of Chapters 1 to 4, we propose direct methods for computing *regular formal solutions* of systems of the form (1) with $m = n$. We also describe a direct approach for computing *k-simple forms* ($k \in \mathbb{N}$) [14, 21, 85] of first-order systems of linear ordinary differential equations. These forms

are useful for the computation of formal solutions without ramification. In the second part, which is formed by Chapters 5 and 6, we are interested in the *reduction algorithms* of linear differential-algebraic equations.

In addition to the theoretical results elaborated in this thesis, another contribution arises in the implementation[1] in MAPLE of most of the algorithms developed and the study of their arithmetic complexity. Finally, it is important to note that the algorithms proposed at least in the first part of this thesis can be generalized to handle linear difference and q-difference systems.

The material of this thesis is organized as follows. Chapter 1 contains classical results on matrix polynomials [50, 65, 42, 104] used in the following chapters. The five remaining chapters contain our contributions. In the sequel, we describe the content of each of them.

In all this thesis, $\mathbb{K}$ denotes a subfield of the field $\mathbb{C}$ of complex numbers and $\overline{\mathbb{K}}$ denotes its algebraic closure.

Chapter 2: Regular Solutions of Higher-Order Linear Differential Systems of the First-Kind

We consider a system of n linear differential equations of order $\ell \geq 1$ of the form

$$\mathcal{L}(x,\vartheta)(y(x)) = A_\ell(x)\,\vartheta^\ell(y(x)) + A_{\ell-1}(x)\,\vartheta^{\ell-1}(y(x)) + \cdots + A_0(x)\,y(x) = 0, \tag{3}$$

where $\vartheta = x\,\frac{d}{dx}$ is the Euler derivation, $A_i(x)$, for $i = 0,\ldots,\ell$, are $n \times n$ matrices of formal power series in x over $\mathbb{K}$ and $y(x)$ is an unknown n-dimensional vector. In this chapter, we assume that $A_\ell(0)$ is invertible and we address the problem of computing the *regular formal solutions* of (3). These latter are linear combinations of solutions of the form

$$y(x) = x^{\lambda_0}\,z(x), \tag{4}$$

where $\lambda_0 \in \overline{\mathbb{K}}$ and $z(x) \in \mathbb{K}[[x]][\log(x)]^n$. Computing such type of solutions is useful in the applications since it can help sometimes in understanding the underlying problem. For example, the existence of logarithm terms in the regular formal solutions of the variational equation of a Hamiltonian system can prove its non-integrability (see [30]).

We will refer to systems of the form (3) satisfying the assumption "$A_\ell(0)$ is invertible" as *systems of the first kind*. Here, the point $x = 0$ is a regular singularity for the system and the space of its formal solutions is spanned by $n\,\ell$ regular solutions.

When $n = 1$ (scalar equations), saying that $A_\ell(0)$ is invertible is equivalent to saying that $A_\ell(0)$ is nonzero. In this case, the exponent λ_0 of any regular solution (4) must be chosen as a root of the so-called *indicial polynomial* which is defined from the equation's coefficients evaluated at $x = 0$. Among the methods computing regular formal solutions of scalar linear ordinary differential equations, we cite the most-known one, the method of Frobenius [44, 37], and the two variants, that of Heffter [53] and that of Poole [86].

When $\ell = 1$ (first-order systems), apart from the methods in [8, 37, 101] specifically dedicated to systems of the first kind, the algorithms developed in [21] and [56, Chap. 9] compute regular solutions of the system even when $x = 0$ is an irregular singularity.

When $\ell \geq 2$, the existing methods for solving directly[2] this problem do not treat the general case, see *e.g.* [5, 64, 79, 80, 81]. They either suppose that the coefficient matrices A_i of System

[1] see `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`

[2] Without the conversion into a first-order system.

(3) are matrix polynomials or solve the problem under some other constraints.

In this chapter, we propose two new direct methods for computing regular formal solutions of systems of the first kind of the form (3) under no assumption.

The first method that we propose is inspired by the work of Poole [86] treating the scalar case. We look for regular formal solutions of the form (4) arranged as series in x whose coefficients are polynomials in $t = \log(x)$ ($\vartheta = \frac{d}{dt}$), that is, of the form

$$y(x) = x^{\lambda_0}(U_0(t) + U_1(t)\,x + \cdots + U_i(t)\,x^i + \cdots)$$

with $\lambda_0 \in \overline{\mathbb{K}}$, $\forall\, m \geq 0$, $U_m(t) \in \overline{\mathbb{K}}[t]^n$ and $U_0 \neq 0$. Plugging $y(x)$ into System (3), one finds that λ_0 and $U_0(t)$ must satisfy

$$L_0(\vartheta)\left(x^{\lambda_0}\,U_0\right) = 0,$$

where $L_0(\lambda)$ is the matrix polynomial defined by

$$L_0(\lambda) = A_\ell(0)\,\lambda^\ell + \cdots + A_1(0)\,\lambda + A_0(0). \tag{5}$$

It follows that λ_0 must be chosen as an eigenvalue of the matrix polynomial $L_0(\lambda)$, *i.e.*, $\det(L_0(\lambda_0)) = 0$, and the coefficients of the polynomial U_0 must form a Jordan chain for $L_0(\lambda)$ associated with λ_0 (see [50]). Hence, one can see that the determinant of $L_0(\lambda)$ will play the same role as the indicial polynomial in the scalar case. For $m \geq 1$, we find

$$L_0(\vartheta + \lambda_0 + m)\,(U_m) = P_m(t),$$

where $P_m(t)$ depends on $U_0(t), \ldots, U_{m-1}(t)$. We have thus reduced the problem of computing regular formal solutions of (3) into two subproblems:

1. Computing regular solutions of linear differential systems with constant coefficients.

2. Computing polynomial solutions in $\log(x)$ of non-homogeneous linear differential systems with constant coefficients.

We explain how to solve these two subproblems by reducing them to solving linear algebraic systems. Thus, we obtain an algorithm which, for a given system of the first kind of the form (3), returns a basis of its formal solution space. We give two variants of this algorithm. In the first variant, each regular solution $x^{\lambda_0}U_0$ in a fundamental system of solutions of system $L_0(\vartheta)(y(x)) = 0$ is extended to form a regular solution of System (3). In the second variant, we gather the eigenvalues of $L_0(\lambda)$ into disjoint sets so that two eigenvalues belonging to two different sets do not differ by integers. Then, for each set of eigenvalues, we compute the general regular formal solution of (3) generated by these eigenvalues. We study the arithmetic complexity of these two variants and present some tables of timings comparing our Maple implementation of this method with two others: the method presented in [5] which reduces the problem of computing regular solutions of systems of the form (3) with polynomial coefficients A_i to the one of computing Laurent series solutions of matrix recurrence equations and the method which consists in converting (3) into first-order system of size $n\,\ell$ then applying the algorithm of [21]. The tables show the efficiency of our algorithm specially on systems of order $\ell \geq 2$.

Another contribution developed in this chapter is the generalization of Frobenius' method [44, 94] to handle systems of the first kind of the form (3). The Frobenius method has been generalized to first-order systems in [92, 54, 56], but the generalizations to higher-order systems found in literature are incomplete, see for instance [64, 80, 81]. Our generalization follows the broad outlines of the Frobenius method in the scalar case and uses classical results on matrix polynomials [50] (notably the notions of partial multiplicities and root polynomials) which

distinguishes it from the other existing generalizations.

Finally, we note that this chapter is a work in collaboration with M. A. Barkatou and T. Cluzeau and is published in [16] and in a part of [17, 18].

Chapter 3: Simple Forms and their Applications in Computing Regular Solutions

This chapter is the subject of the published paper [18] in collaboration with M. A. Barkatou and T. Cluzeau.

We consider here systems of the same form as (3) except that $A_\ell(0)$ and $A_\ell(x)$ are not necessarily supposed to be invertible, *i.e*, arbitrary matrices. With such systems, we associate the matrix polynomial $L_0(\lambda)$ defined in (5).

We first examine the case where the matrix polynomial $L_0(\lambda)$ is regular, *i.e.*, $\det(L_0(\lambda)) \neq 0$. We will call a system of the form (3) (or the corresponding operator $\mathcal{L}(x, \vartheta)$) having a regular matrix polynomial $L_0(\lambda)$ a *simple system* (or a *simple operator*). The notion of simple systems has been first introduced by Barkatou in computing rational solutions of first-order systems of linear ODEs (see [14]). Then, it has been used by Barkatou & Pflügel [21] in the calculation of regular solutions (always for first-order systems). These systems are called so because their indicial polynomials can be easily computed.

Remark that a system of the first kind is necessarily simple but the converse is not always true. We also point out that the class of simple systems may include linear differential-algebraic equations, since the leading coefficient matrix $A_\ell(x)$ is not supposed to be invertible.

We show in this chapter that the methods developed in Chapter 2 for computing regular formal solutions of systems of the first kind are still valid for simple systems and that

> *The dimension of the $\overline{\mathbb{K}}$-vector space spanned by the regular formal solutions of a simple linear differential system of the form (3) is equal to the degree of the determinant of $L_0(\lambda)$.*

This allows us to deduce the nature of the singularity $x = 0$:

> *A simple linear differential system of the form (3) with invertible leading coefficient matrix $A_\ell(x)$ over the field of Laurent series $\mathbb{K}((x))$ has a regular singularity at the point $x = 0$ if and only if System (3) is of the first kind.*

Another contribution of this chapter is the study of the arithmetic complexity of the algorithm presented in [21] for computing the regular solutions of simple systems of first-order. The purpose of this study is to compare, from an arithmetic complexity point of view, our approach for solving directly simple systems of the form (3) with the one that consists in transforming System (3) into a first-order simple system of size $n\,\ell$ then using the algorithm of [21].

Then, we investigate non-simple systems, *i.e.*, systems of the form (3) for which the matrix polynomial $L_0(\lambda)$ is singular ($\det(L_0(\lambda)) = 0$). As our methods for computing regular solutions are no longer applicable, we propose to compute a simple linear differential system $\overline{\mathcal{L}}(x, \vartheta)(z(x)) = 0$ from which one can get the regular formal solutions of the non-simple one. To achieve this, we need to suppose that the leading coefficient matrix $A_\ell(x)$ of System (3) is invertible over $\mathbb{K}((x))$, *i.e.*, we only consider explicit differential systems, to guarantee that the regular formal solution space of (3) is of finite dimension.

The problem of computing a simple system, from which one can recover the solutions of the non-simple system $\mathcal{L}(x,\vartheta)(y(x)) = 0$, has been already studied in [21] (see also [14]) for the case $\ell = 1$: Barkatou & Pflügel [21] showed that, using the super-reduction algorithm [24, 25, 58], one can construct two matrices $S(x)$ and $T(x)$ invertible in $\mathbb{K}((x))^{n\times n}$ such that the operator $\overline{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ is simple. Note that the formal solutions space of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ and that of $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ are then isomorphic since $y(x)$ and $z(x)$ are related by $y(x) = T(x)\,z(x)$ and $T(x)$ is invertible.

Unfortunately, when dealing with higher-order systems ($\ell \geq 2$), it is not always possible to find two invertible matrices $S(x)$ and $T(x)$ such that the operator $S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ is simple[3]. Indeed, we show that, given a non-simple system with $n = \ell = 2$, if a right minimal basis and a left minimal basis [65, 42] of the associated matrix polynomial $L_0(\lambda)$ do not contain nonzero constant vectors then for any matrices $S(x)$ and $T(x)$ in $\mathbb{K}((x))^{2\times 2}$, the operator $S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ is always non-simple. For this reason, we are first interested in the existence of a linear substitution $y(x) = T(x)\,z(x)$ with invertible matrix $T(x)$ such that the linear differential system satisfied by $z(x)$ is simple. We give a necessary condition for the existence of a such linear substitution:

> *Given a non-simple explicit differential system of the form (3), if there exists an invertible matrix $T(x) \in \mathbb{K}((x))^{n\times n}$ such that the system $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$, where $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T(x)$, is simple, then the elements of a right minimal basis of the matrix polynomial $L_0(\lambda)$ associated with (3) are all constant vectors.*

As a consequence, we propose an algorithm that either decides the existence of such a linear substitution and computes it, or proves that it does not exist. In the latter case, we give a differential variant of the *EG'-algorithm* developed by Abramov, Bronstein & Khmelnov in [4, Section 4]. The EG'-algorithm, an improved version of the EG-algorithm described in [1], has been elaborated in order to bring a matrix recurrence equation into another one whose leading (or trailing) coefficient matrix is nonsingular. In this chapter, we adapt this algorithm to our case: we have a linear differential system with singular matrix polynomial $L_0(\lambda)$ and we would like to compute another system the associated matrix polynomial of which is regular. To achieve this, we need to suppose that the non-simple system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ has polynomial coefficients. This algorithm applies a series of elementary operations to the rows of the operator $\mathcal{L}(x,\vartheta)$ and always yields a simple system from which the regular solutions of the original system can be recovered. Note that, the obtained simple system could be of order greater than ℓ and it is not necessarily equivalent to the original system in the sense that the formal solution spaces of the two systems may not be isomorphic; when they are not, we explain how regular solutions of the non-simple system can be obtained.

Another contribution of this chapter is the MAPLE implementation and the arithmetic complexity study of the new algorithms presented.

Chapter 4: On k-Simple Forms of First-Order Linear Differential Systems and their Computation

The idea behind this chapter comes from the work [85] of Pflügel.

Consider a system of linear ordinary differential equations of first-order of the form

$$\vartheta(y(x)) = A(x)\,y(x) \quad \text{with } A(x) = \frac{1}{x^p}\left(A_0 + A_1\,x + \cdots + A_k\,x^k + \cdots\right), \tag{6}$$

[3] One must introduce ϑ in S and T.

where $p \in \mathbb{N}$ and the A_i's are $n \times n$ matrices with entries in $\mathbb{K}$ such that $A_0 \neq 0$. In [85], in order to compute the formal solutions of (6), the author is led to consider systems in a more general form than (6). Indeed, for an integer $k \in \{0, \ldots, p-1\}$, define the matrix $D(x) = \mathrm{diag}(x^{\alpha_1}, \ldots, x^{\alpha_n})$ with $\alpha_i = \max\{0, -k - v(A(x)(i,.))\}$, where $A(x)(i,.)$ denotes the ith row of matrix $A(x)$. Multiplying System (6) on the left by $x^k D(x)$, one obtains

$$\mathcal{D}_k(y(x)) = D(x)\,\vartheta_k(y(x)) + N(x)y(x) = 0, \tag{7}$$

where $\vartheta_k = x^k\,\vartheta$ and $N(x) = -x^k D(x)A(x)$ has formal power series entries.

Studying the action of $\mathcal{D}_k$ on a solution of the form $y = \exp\left(\int w\right) z$ with

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots \in \overline{\mathbb{K}((x))},$$

and $z \in \overline{\mathbb{K}}[[x^{\frac{1}{r}}]]^n$ $(r \in \mathbb{N}^*)$ such that $z(0) \neq 0$, one finds

$$\exp\left(-\int w\right) \mathcal{D}_k(y) = \left(D(0)\,\lambda_0 + N(0)\right) z(0) + \cdots,$$

where the dots stand for terms of higher valuation. It follows that if $y = \exp\left(\int w\right) z$ is a solution of $\mathcal{D}_k(y) = 0$ then $\left(D(0)\,\lambda_0 + N(0)\right) z(0) = 0$ which implies the two following conditions

$$\det(D(0)\,\lambda_0 + N(0)) = 0 \quad \text{and} \quad z(0) \in \ker(D(0)\,\lambda_0 + N(0)).$$

Hence, it is natural to expect that the roots of the determinant of the matrix pencil

$$L_k(\lambda) = D(0)\,\lambda + N(0)$$

will play an important role in the determination of the formal solutions. But it may happen that the determinant $\det(L_k(\lambda))$ vanishes identically in λ in which case, it is quite useless. Pflügel called systems of the form (7) with a regular matrix pencil $L_k(\lambda)$ *k-simple systems*[4] and showed in [85, Th. 3.3] that if System (6) can be written as a k-simple system of the form (7) and if λ_0 is an eigenvalue of the matrix pencil $L_k(\lambda)$ of algebraic multiplicity m, then there exist m linearly independent formal solutions of (6) of the form $y = \exp\left(\int w\right) z$, where $w \in x^{-1}\overline{\mathbb{K}}[x^{-1/r}]$ (for some $r \in \mathbb{N}^*$) can be written as

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots,$$

where the dots stand for terms of higher valuation, and $z \in \overline{\mathbb{K}}[[x^{1/r}]]^n[\log(x)]$. Hence, the determinant of $L_k(\lambda)$ plays the same role as the Newton polynomials (or characteristic polynomials) in the scalar case.

Remark that a system of the form (6) written as in (7) is not always k-simple. However, while studying the super-redution algorithm (see [58, 24, 25]), the author of [85] noticed that the polynomials defined from a super-reduced form[5] are strongly connected to the Newton polynomials. He showed that if System (6) is super-reduced, then it can be written as a k-simple system for $k = 0, \ldots, p-1$.

In this chapter, we will consider the class of systems of the form (7) with $k \in \mathbb{N}$, $D(x)$ and $N(x)$ in $\mathbb{K}[[x]]^{n \times n}$ such that $D(x)$ is invertible over $\mathbb{K}((x))$ ($D(x)$ is not necessarily supposed to be diagonal). Before this thesis, the only method to compute a k-simple system equivalent

[4] We will also call them *systems simple with respect to* ϑ_k. The notion of k-simplicity comes as a generalization of that of simplicity viewed in [14, 21].

[5] The super-reduced forms are defined for systems of the form (6).

to (7) was by using the super-reduction algorithm. But, a k-simple system of the form (7), written as in (6) with $A(x) = -x^{-k}\, D^{-1}(x)\, N(x)$, is not necessarily super-reduced. Hence, we develop in this chapter a direct algorithm that brings System (7) into an equivalent k-simple one $\widetilde{\mathcal{D}}_k(z(x)) = \widetilde{D}(x)\, \vartheta_k(z(x)) + \widetilde{N}(x)\, z(x) = 0$. Our approach is based on the algebraic treatment of the matrix pencil $L_k(\lambda)$ and proceeds in a similar way as in [58]. Finally, we point out three characteristics of our algorithm :

- The k-simple system produced by our approach is not necessarily super-reduced.
- Our algorithm applied to $\mathcal{D}_k$ given in (7) with $k = 0$ allows us to determine the nature of the singularity $x = 0$. It can then be considered as an alternative way of Moser's reduction [77] to determine the nature of the singularity $x = 0$.
- Our approach applied to $\mathcal{D}_k$ given in (7) preserves the simplicity with respect to ϑ_{k+i} for $i = 1, 2, \ldots$. In other words, if the operator $\mathcal{D}_k$ written respectively with ϑ_{k+i} for $i = 1, 2, \ldots$ is simple with respect to ϑ_{k+i}, then after applying our algorithm to $\mathcal{D}_k$, the returned operator is also simple with respect to ϑ_{k+i} for $i = 1, 2, \ldots$.

Here again, we study the arithmetic complexity of our algorithm which has been implemented[6] in MAPLE and we clarify our approach using examples.

Chapter 5: Reduction Algorithms for Linear Differential-Algebraic Equations of First-Order

The content of this chapter constitutes a part of the published paper [20] in collaboration with M. A. Barkatou and E. Pflügel.

We study linear differential-algebraic equations of first-order of the form

$$L(y(x)) = A(x)\, \partial(y(x)) + B(x)\, y(x) = f(x), \tag{8}$$

where $\partial = \frac{d}{dx}$, x is a complex variable, $A(x)$ and $B(x)$ are $m \times n$ matrices of formal power series over $\mathbb{K}$, $y(x)$ is an unknown n-dimensional vector and the right-hand side $f(x)$ is an m-dimensional vector of formal power series over $\mathbb{K}$. The purpose of our study is to generalize the concepts developed for first-order systems of ODEs to the DAEs case. To achieve this, the first idea that comes to mind is to reduce DAEs of the form (8) to a standard form which can be handled via well known techniques as, for example, those elaborated for ODEs. In this chapter, we develop a new approach decoupling the DAE (8) into an explicit differential system of first-order and a purely algebraic one (and evidently some necessary conditions on the right-hand side).

Regardless of their goals, most of the symbolic and numerical algorithms manipulating linear DAEs of first-order (see [47, 67, 88]) follow the same techniques:

- multiply the DAEs on the left by an invertible matrix,
- differentiate the algebraic equations,
- make the change of variables $y(x) = T(x)\, z(x)$ with invertible matrix $T(x)$.

[6] The code is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`

These operations are not always enough to entirely decouple a DAE of the form (8) into a first-order system of ODEs and a system of algebraic equations. However, having a closer look on the algorithm of Harris, Sibuya & Weinberg [52], we have noticed that by authorizing also change of variables of the form $y(x) = T(z(x))$, where T is a unimodular matrix differential operator, one can achieve decoupling.

In this chapter, we first review the algorithm of Harris et al. [52]; we use a new presentation by expressing the operations performed as a series of left- and right- (differential) transformations on the matrix differential operator L. Then, inspired by the latter algorithm, we give a new and simplified approach to show that

> *Given a matrix differential operator $L = A(x)\,\partial + B(x)$, there exist two unimodular matrix differential operators S and T such that $\widetilde{L} = S\,L\,T$ is of the form*
>
> $$\widetilde{L} = \begin{pmatrix} \widetilde{A}_{11}(x)\,\partial + \widetilde{B}_{11}(x) & 0 & 0 \\ 0 & \widetilde{B}_{22}(x) & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{9}$$
>
> *where $\widetilde{A}_{11}$ and $\widetilde{B}_{22}$ are both invertible.*

Therefore, the DAE (8) is reduced to two separated systems:

1. the first-order system of linear ODEs $\widetilde{A}_{11}(x)\,z_1'(x) + \widetilde{B}_{11}(x)\,z_1(x) = \widetilde{f}_1(x)$, and
2. the system of algebraic equations $\widetilde{B}_{22}(x)\,z_2(x) = \widetilde{f}_2(x)$,

together with some necessary conditions on the right-hand side expressed by $\widetilde{f}_3(x) = 0$. Here, we have

$$y(x) = T\left(\begin{pmatrix} z_1(x) \\ z_2(x) \\ z_3(x) \end{pmatrix}\right) \quad \text{and} \quad \begin{pmatrix} \widetilde{f}_1(x) \\ \widetilde{f}_2(x) \\ \widetilde{f}_3(x) \end{pmatrix} = S(f(x)).$$

Thus, bringing the DAE to an ODE problem, it is then possible to extend the notion of regular and irregular singularities to linear DAEs of first-order.

To get the decoupled form (9), we proceed as follows. We start by applying the row-reduction procedure [27] on the input matrix differential operator L. This is translated by a left-multiplication of L by a unimodular matrix differential operator. Then, we follow it by a column-reduction on the resulting row-reduced operator (right-multiplication by a unimodular matrix differential operator). The obtained operator is now column-reduced but not necessarily row-reduced. If it is not so, then we apply again row-reduction and so on, until we end up with an operator which is row-reduced and column-reduced. We show that this happens after a finite number of iterations. Finally, by means of multiplications by invertible matrices, we get an operator of the form (9).

Chapter 6: On Simultaneous Row and Column Reduction of Higher-Order Linear Differential Systems

This chapter is the subject of the paper [19] in collaboration with M. A. Barkatou, G. Labahn and E. Pflügel.

In this chapter, we generalize the method proposed in Chapter 5 for first-order linear DAEs to higher-order ones of the form (1). We act on the matrix differential operator L defining System (1), *i.e.*,

$$L = \mathcal{A}_\ell(x)\,\partial^\ell + \mathcal{A}_{\ell-1}(x)\,\partial^{\ell-1} + \cdots + \mathcal{A}_0(x),$$

in order to obtain an equivalent operator which is *simultaneously row and column reduced*. This latter is of order less than or equal to that of L and has a particular block structure allowing to decouple the associated system into a purely algebraic system, a purely differential system (square and its equations are linearly independent) and some necessary conditions on the right-hand side. Furthermore, the obtained purely differential system can be easily rewritten as an explicit first-order differential system and hence the dimension of its formal solution space can be determined and the classification of singularities can be explored.

To compute a simultaneously row and column reduced form, we propose two approaches. The first approach consists in applying alternatively row-reduction and column-reduction until we find an operator which is at the same time row-reduced and column-reduced. Although this method works on rows and columns, the final operator has a structure by blocks. The second approach is inspired by techniques used for computing Popov forms in the commutative case (see, *e.g.*, [78]) except that here we manipulate blocks.

Finally, we note that a simultaneously row and column reduced form is weaker than the Popov and Jacobson normal forms.

Part I

Simple Forms and Regular Formal Solutions of Systems of Linear Differential Equations

Chapter 1

A Survey on Matrix Polynomials

Contents

Let $\mathbb{K}$ be a subfield of the field $\mathbb{C}$ of complex numbers ($\mathbb{Q} \subseteq \mathbb{K} \subseteq \mathbb{C}$) and $\overline{\mathbb{K}}$ its algebraic closure. An $n \times n$ *matrix polynomial* is an $n \times n$ matrix whose entries are polynomials in $\mathbb{K}[\lambda]$. Since $\mathbb{K}[\lambda]^{n\times n}$ is isomorphic to $\mathbb{K}^{n\times n}[\lambda]$, any matrix polynomial $L(\lambda)$ can be written in the form

$$L(\lambda) = A_\ell\,\lambda^\ell + \cdots + A_1\,\lambda + A_0, \tag{1.1}$$

where $\ell \in \mathbb{N}$, $A_0, A_1, \ldots, A_\ell$ are $(\ell+1)$ matrices of $\mathbb{K}^{n\times n}$ and $A_\ell \neq 0$. Hence, $L(\lambda)$ is said to be *of degree* ℓ and the matrix A_ℓ is called the *leading coefficient matrix* of $L(\lambda)$. Matrix polynomials of degree one are usually known as *matrix pencils*. The theory of matrix polynomials has been extensively studied in literature: see, *e.g.*, [45, 50, 65, 69]. This chapter aims at recalling some basic results on matrix polynomials such as the Smith normal form, partial multiplicities, Jordan chains, minimal bases, etc. We will use these mathematical objects in the study of linear differential systems which is the main topic of this thesis. For the computation of partial multiplicities and Jordan chains, which are useful in the algorithms developed in the next chapters, we review an algorithm described in [104] (see also [97]) and provide here its arithmetic complexity analysis.

The chapter is divided into 7 sections. Sections 1.1 to 1.6 are devoted to regular matrix polynomials whereas Section 1.7 concerns only singular matrix polynomials.

1.1 Regular matrix polynomials

The *rank* of a matrix polynomial $L(\lambda)$ is defined as

$$\operatorname{rank}(L(\lambda)) = \max\left\{\operatorname{rank}(L(\lambda_0))\,;\ \lambda_0 \in \overline{\mathbb{K}}\right\}.$$

It is also the rank of $L(\lambda)$ viewed as a matrix with entries in the field $\mathbb{K}(\lambda)$ of rational functions in one variable λ.

Definition 1.1.1. *An $n \times n$ matrix polynomial $L(\lambda)$ is said to be* regular *if* $\operatorname{rank}(L(\lambda)) = n$*, or, equivalently, if its determinant* $\det(L(\lambda))$ *does not vanish identically. Otherwise, it is said to be* singular.

The degree of the determinant of a regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ of degree ℓ is less than or equal to $n\,\ell$ and the equality holds if and only if the leading coefficient matrix of $L(\lambda)$ is invertible; this can be shown by considering the "reverse" polynomial $\lambda^\ell L(\lambda^{-1})$ of $L(\lambda)$.

Example 1.1.1. *Consider the* 2×2 *matrix polynomial of degree* 6 *given by*

$$L(\lambda) = \begin{pmatrix} \lambda^2 - 2\,\lambda + 1 & -2\,\lambda^5 + 2\,\lambda^4 + 3\,\lambda^3 - 4\,\lambda^2 + \lambda \\ -2\,\lambda^3 + 5\,\lambda^2 - 4\,\lambda + 1 & 2\,\lambda^6 - 8\,\lambda^4 + 7\,\lambda^3 - \lambda \end{pmatrix} \tag{1.2}$$

$$= \begin{pmatrix} (\lambda-1)^2 & -\lambda\left(2\,\lambda^2 + 2\,\lambda - 1\right)(\lambda-1)^2 \\ (1-2\,\lambda)\,(\lambda-1)^2 & \lambda\left(2\,\lambda^3 + 4\,\lambda^2 - 2\,\lambda - 1\right)(\lambda-1)^2 \end{pmatrix}.$$

Its determinant

$$\det(L(\lambda)) = \lambda\,(\lambda-1)^4 \begin{vmatrix} 1 & -(2\,\lambda^2 + 2\,\lambda - 1) \\ 1 - 2\,\lambda & 2\,\lambda^3 + 4\,\lambda^2 - 2\,\lambda - 1 \end{vmatrix} = -2\,\lambda\,(\lambda+1)\,(\lambda-1)^6 \tag{1.3}$$

is a nonzero polynomial hence $L(\lambda)$ *is a regular matrix polynomial. Remark that the degree of* $\det(L(\lambda))$ *is equal to* 8*, less than* 2×6*, which is coherent with the fact that the leading coefficient matrix of* $L(\lambda)$

$$A_6 = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$$

is singular. ■

Definition 1.1.2. *Let* $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ *be a regular matrix polynomial.*

1. *An element* λ_0 *of* $\overline{\mathbb{K}}$ *is called an* eigenvalue *(or a* finite eigenvalue*) of* $L(\lambda)$ *if* λ_0 *is a root of the determinant of* $L(\lambda)$*,* i.e.*,* $\det(L(\lambda_0)) = 0$*. If* $\lambda_0 \in \overline{\mathbb{K}}$ *is an eigenvalue of* $L(\lambda)$*, then its multiplicity as a root of* $\det(L(\lambda))$ *is called the* algebraic multiplicity *of* λ_0 *and it is denoted by* $m_a(\lambda_0)$*.*

2. *The set of all eigenvalues of* $L(\lambda)$ *is called the* spectrum *of* $L(\lambda)$ *and denoted by* $\sigma(L)$ *(*$\sigma(L) \subset \overline{\mathbb{K}}$*).*

3. *For* $\lambda_0 \in \sigma(L)$*, a nonzero vector* v *of* $\overline{\mathbb{K}}^n$ *is called an* eigenvector *of* $L(\lambda)$ *associated with* λ_0 *if* v *belongs to the right nullspace* $\ker(L(\lambda_0))$ *of* $L(\lambda_0)$*,* i.e.*,* $L(\lambda_0)\,v = 0$*. The dimension of* $\ker(L(\lambda_0))$ *is called the* geometric multiplicity *of* λ_0 *and it is denoted by* $m_g(\lambda_0)$*.*

This definition of eigenvalues of regular matrix polynomials is a generalization of the well-known notion of eigenvalues of square constant matrices. Indeed, an eigenvalue of a matrix $A \in \mathbb{K}^{n\times n}$ is, by definition, an eigenvalue of the regular matrix pencil $I_n\lambda - A$.

We illustrate these notions with the following example.

Example 1.1.2. *Consider the matrix polynomial* $L(\lambda)$ *given by (1.2). From (1.3), we can see that* $L(\lambda)$ *has three eigenvalues* -1*,* 0 *and* 1 *of algebraic multiplicity* 1*,* 1 *and* 6 *respectively. To compute their geometric multiplicities, we evaluate* $L(\lambda)$ *respectively at* $\lambda_0 = -1$*,* 0 *and* 1*. We obtain*

$$L(-1) = \begin{pmatrix} 4 & -4 \\ 12 & -12 \end{pmatrix}, \quad L(0) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \textit{and} \quad L(1) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Thus, $m_g(-1) = 1$*,* $m_g(0) = 1$ *and* $m_g(1) = 2$*. An eigenvector respectively associated with the eigenvalue* $\lambda_0 = -1$*,* 0 *and* 1 *is any vector of* $\mathbb{C}^2$ *respectively of the form*

$$\begin{pmatrix} a \\ a \end{pmatrix}, \quad \begin{pmatrix} 0 \\ b \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} c \\ d \end{pmatrix},$$

where $a, b, c, d \in \mathbb{C}$ *such that* $a \neq 0$*,* $b \neq 0$ *and* $(c, d) \neq (0, 0)$*.* ■

Remark 1.1.1. In the algorithms that we develop in the next chapters, we need to compute a representation of the spectrum of a regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ of degree ℓ. We proceed as follows. We compute the determinant of $L(\lambda)$, then we factor it over $\mathbb{K}$. Thus, each eigenvalue $\lambda_0 \in \sigma(L)$ is represented by RootOf($p(\lambda)$) (using MAPLE notation) where $p(\lambda)$ is an irreducible factor of $\det(L(\lambda))$ in $\mathbb{K}[\lambda]$ with $p(\lambda_0) = 0$. Note that λ_0 belongs to $\mathbb{K}$ if and only if the polynomial $p(\lambda)$ is of degree 1. Randomized algorithms allow us to compute the determinant of $L(\lambda)$ in $O^{\sim}(n^{\omega}\,\ell)$ operations in $\mathbb{K}$ (see [62]), where $2 \leq \omega < 2.376$ is the exponent for the complexity of matrix multiplication (see [100]), and to factor it over $\mathbb{K}$ in $O^{\sim}\left((n\,\ell)^{12}\right)$ operations in $\mathbb{K}$ (see [76, Algorithm 18.7.3]). In the sequel, the cost of computing $\sigma(L)$ will not be taken into account in the complexity analysis of our algorithms.

1.2 The Smith normal form and partial multiplicities

Definition 1.2.1. *Two matrix polynomials $A(\lambda)$ and $B(\lambda)$ of $\mathbb{K}[\lambda]^{n\times n}$ are said to be* equivalent *if there exist two unimodular matrix polynomials $E(\lambda)$ and $F(\lambda)$ of $\mathbb{K}[\lambda]^{n\times n}$,* i.e., *with constant nonzero determinant in $\mathbb{K}$, such that*

$$A(\lambda) = E(\lambda)\,B(\lambda)\,F(\lambda). \tag{1.4}$$

In the sequel, we denote this equivalence relation by the symbol $\backsim$, *i.e.*, we write $A(\lambda) \backsim B(\lambda)$ if $A(\lambda)$ is equivalent to $B(\lambda)$.

Notice that if $A(\lambda) \backsim B(\lambda)$, then $A(\lambda)$ is regular if and only if $B(\lambda)$ is regular and we have $\sigma(A) = \sigma(B)$.

Since the ring $\mathbb{K}[\lambda]$ is a principal ideal domain, any matrix polynomial $L(\lambda)$ admits a *Smith normal form*, *i.e.*, a diagonal representation, as it is stated by the following theorem.

Theorem 1.2.1 ([50], Th. S1.1)**.** *Any regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ is equivalent to a unique matrix polynomial of the form*

$$S(\lambda) = \operatorname{diag}(d_1(\lambda), \ldots, d_n(\lambda))\,, \tag{1.5}$$

where the d_i's are monic polynomials and $d_i(\lambda)$ divides $d_{i+1}(\lambda)$, for $i = 1, \ldots, n-1$.

Definition 1.2.2. *The matrix polynomial $S(\lambda)$ given by (1.5) is called the* Smith normal form *of $L(\lambda)$ and the monic polynomials $d_i(\lambda)$ are called the* invariant polynomials of $L(\lambda)$.

The Smith normal form of a matrix polynomial $L(\lambda)$ can be obtained by applying a sequence of elementary row and column operations to $L(\lambda)$. Randomized algorithms allow to compute the Smith normal form of an $n \times n$ matrix polynomial of degree ℓ in at most $O^{\sim}(n^{\omega}\,\ell)$ operations in $\mathbb{K}$ (see [62, 93] and references therein).

Example 1.2.1. *Using for example the command* SmithForm *of the* LINEARALGEBRA *package of* MAPLE*, we can compute the Smith normal form $S(\lambda)$ of the matrix polynomial $L(\lambda)$ given by (1.2) together with two unimodular matrix polynomials $E(\lambda)$ and $F(\lambda)$ providing the equivalence $L(\lambda) \backsim S(\lambda)$. We find*

$$\begin{aligned} S(\lambda) &= \begin{pmatrix} \lambda^2 - 2\,\lambda + 1 & 0 \\ 0 & \lambda^6 - 3\,\lambda^5 + 2\,\lambda^4 + 2\,\lambda^3 - 3\,\lambda^2 + \lambda \end{pmatrix} \\ &= \begin{pmatrix} (\lambda-1)^2 & 0 \\ 0 & \lambda\,(\lambda+1)\,(\lambda-1)^4 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & 0 \\ -\lambda^4 + \lambda^3 + \lambda^2 - 2\,\lambda + \frac{1}{2} & -\frac{1}{2} \end{pmatrix}}_{E(\lambda)} L(\lambda) \underbrace{\begin{pmatrix} 2\,\lambda^3 + 2\,\lambda^2 - \lambda + 1 & 2\,\lambda^3 + 2\,\lambda^2 - \lambda \\ 1 & 1 \end{pmatrix}}_{F(\lambda)}, \end{aligned} \tag{1.6}$$

with $\det(E(\lambda)) = -\frac{1}{2}$ *and* $\det(F(\lambda)) = 1$. ■

Due to the equivalence, each invariant polynomial $d_i(\lambda)$ of positive degree can be written as

$$d_i(\lambda) = \prod_{j=1}^{N_i} (\lambda - \lambda_j)^{\alpha_{i,j}} ,$$

where $\lambda_j \in \sigma(L)$ such that $\lambda_i \neq \lambda_j$ for $i \neq j$ and $\alpha_{i,j}, N_i \in \mathbb{N}^*$ such that $N_i \leq N_{i+1}$ and $\alpha_{i,j} \leq \alpha_{i+1,j}$ (since $d_i(\lambda)$ divides $d_{i+1}(\lambda)$). The factors $(\lambda - \lambda_j)^{\alpha_{i,j}}$ are called the *elementary divisors* of $L(\lambda)$.

Theorem 1.2.2 ([50], Th. S1.10)**.** *Let* $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ *be a regular matrix polynomial and* $\lambda_0 \in \sigma(L)$. *There exist two matrix polynomials* $E_{\lambda_0}(\lambda)$ *and* $F_{\lambda_0}(\lambda)$ *of* $\overline{\mathbb{K}}[\lambda]^{n\times n}$ *invertible at* $\lambda = \lambda_0$, i.e., $E_{\lambda_0}(\lambda_0), F_{\lambda_0}(\lambda_0) \in \mathrm{GL}_n(\overline{\mathbb{K}})$, *such that*

$$L(\lambda) = E_{\lambda_0}(\lambda)\, S_{\lambda_0}(\lambda)\, F_{\lambda_0}(\lambda)$$

with $S_{\lambda_0}(\lambda)$ *of the form*

$$S_{\lambda_0}(\lambda) = \mathrm{diag}(1, \ldots, 1, (\lambda - \lambda_0)^{\kappa_1}, \ldots, (\lambda - \lambda_0)^{\kappa_{m_g(\lambda_0)}}) , \tag{1.7}$$

where the κ_i*'s are positive integers uniquely determined from* $L(\lambda)$ *and* λ_0, *satisfying* $\kappa_1 \leq \cdots \leq \kappa_{m_g(\lambda_0)}$ *and* $\sum_{i=1}^{m_g(\lambda_0)} \kappa_i = m_a(\lambda_0)$.

Definition 1.2.3. *The matrix polynomial* $S_{\lambda_0}(\lambda)$ *given by (1.7) is called the* local Smith form of $L(\lambda)$ at λ_0 *and the integers* κ_i, *for* $i = 1, \ldots, m_g(\lambda_0)$, *are called the* partial multiplicities of $L(\lambda)$ at λ_0.

Example 1.2.2. *From the Smith normal form (1.6) of the matrix polynomial* $L(\lambda)$ *defined by (1.2), we can get the local Smith form of* $L(\lambda)$ *at* $\lambda_0 = 1$ *as follows. We have*

$$\begin{aligned} L(\lambda) &= E^{-1}(\lambda)\, S(\lambda)\, F^{-1}(\lambda) \\ &= E^{-1}(\lambda) \begin{pmatrix} 1 & 0 \\ 0 & \lambda\,(\lambda+1) \end{pmatrix} \begin{pmatrix} (\lambda-1)^2 & 0 \\ 0 & (\lambda-1)^4 \end{pmatrix} F^{-1}(\lambda) \\ &= E_1(\lambda)\, S_1(\lambda)\, F_1(\lambda), \end{aligned}$$

where

$$E_1(\lambda) = E^{-1}(\lambda) \begin{pmatrix} 1 & 0 \\ 0 & \lambda\,(\lambda+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -2\,\lambda^4 + 2\,\lambda^3 + 2\,\lambda^2 - 4\,\lambda + 1 & -2\,\lambda\,(\lambda+1) \end{pmatrix},$$

$$F_1(\lambda) = F^{-1}(\lambda) = \begin{pmatrix} 1 & -2\,\lambda^3 - 2\,\lambda^2 + \lambda \\ -1 & 2\,\lambda^3 + 2\,\lambda^2 - \lambda + 1 \end{pmatrix},$$

and

$$S_1(\lambda) = \begin{pmatrix} (\lambda-1)^2 & 0 \\ 0 & (\lambda-1)^4 \end{pmatrix}$$

is the local Smith form of $L(\lambda)$ *at* $\lambda_0 = 1$. *Here, the eigenvalue* 1 *has* 2 *(*$2 = m_g(1)$*) partial multiplicities* $\kappa_1 = 2$ *and* $\kappa_2 = 4$ *with* $\kappa_1 + \kappa_2 = m_a(1) = 6$. ■

The usefulness of the Smith normal form is twofold. On one hand, it allows to define the partial multiplicities at a given eigenvalue which make the difference between scalar polynomials ($n = 1$) and matrix polynomials ($n > 1$). Indeed, in the scalar case ($n = 1$), an eigenvalue of a polynomial $L(\lambda)$, which is a root of $L(\lambda)$, has a unique partial multiplicity equal to its (algebraic) multiplicity. This is not always true in the matrix case ($n > 1$) as shown in Example 1.2.2 where the algebraic multiplicity of the eigenvalue 1 is split into two partial multiplicities. Additionally, the partial multiplicities determine the maximum lengths of Jordan chains that we will consider later. On the other hand, due to its diagonal form, it is sometimes easier to work with the Smith normal form of a matrix polynomial $L(\lambda)$ instead of working with $L(\lambda)$ itself.

We will explain later, in Section 1.6, how one can obtain the partial multiplicities of $L(\lambda)$ at a given eigenvalue λ_0 without computing the local Smith form of $L(\lambda)$ at λ_0.

1.3 Linearization

From a theoretical point of view, it is sometimes easier to reduce a problem of higher degree to another problem of degree one. For matrix polynomials of degree greater than or equal to 2, this can be done by computing an "equivalent" matrix pencil called a *linearization*.

Definition 1.3.1. *Let $L(\lambda)$ be an $n \times n$ matrix polynomial of degree ℓ. A matrix pencil $P(\lambda) = P_1\,\lambda + P_0$, where $P_1, P_0 \in \mathbb{K}^{n\ell \times n\ell}$, is called a* linearization *of $L(\lambda)$ if*

$$P(\lambda) \backsim \begin{pmatrix} L(\lambda) & 0 \\ 0 & I_{n(\ell-1)} \end{pmatrix}.$$

Computing a linearization of a regular matrix polynomial $L(\lambda)$ is one of the classical approaches for investigating the spectral information of $L(\lambda)$ provided by the elementary divisors. Indeed, from Definitions 1.2.1 and 1.3.1, it derives that $L(\lambda)$ and its linearization $P(\lambda)$ share the same spectrum. Moreover, if $S(\lambda)$ denotes the Smith normal form of $L(\lambda)$ then

$$P(\lambda) \backsim \begin{pmatrix} L(\lambda) & 0 \\ 0 & I_{n(\ell-1)} \end{pmatrix} \backsim \begin{pmatrix} S(\lambda) & 0 \\ 0 & I_{n(\ell-1)} \end{pmatrix} \backsim \begin{pmatrix} I_{n(\ell-1)} & 0 \\ 0 & S(\lambda) \end{pmatrix}.$$

The matrix polynomial $\operatorname{diag}\big(I_{n(\ell-1)}, S(\lambda)\big)$ is a diagonal matrix whose ith diagonal entry is monic and divides the $(i+1)$th one. From the uniqueness of the Smith normal form, it follows that $\operatorname{diag}\big(I_{n(\ell-1)}, S(\lambda)\big)$ is the Smith normal form of $P(\lambda)$. Thus, the elementary divisors of $L(\lambda)$ and of $P(\lambda)$ are the same.

Theorem 1.3.1 ([50], page 186)**.** *Let $L(\lambda) = \sum_{i=0}^{\ell} A_i\,\lambda^i$ be an $n \times n$ matrix polynomial of degree ℓ. The matrix pencil defined by*

$$\mathcal{C}_L(\lambda) = \begin{pmatrix} I_n & 0 & 0 & \cdots & 0 \\ 0 & I_n & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \ddots & I_n & 0 \\ 0 & \cdots & \cdots & 0 & A_\ell \end{pmatrix} \lambda + \begin{pmatrix} 0 & -I_n & 0 & \cdots & 0 \\ 0 & 0 & -I_n & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & -I_n \\ A_0 & A_1 & \cdots & \cdots & A_{\ell-1} \end{pmatrix} \tag{1.8}$$

and called the first companion form *(or* companion polynomial*) of $L(\lambda)$ is a linearization of $L(\lambda)$.*

The general point of view of this thesis is to avoid increasing the problem dimension. Thus, we will merely use the notion of linearization for theoretical purposes and not for practical ones. Therefore, we do not provide more information on this notion here and we invite the reader to consult the papers [68, 72] for additional details.

Remark 1.3.1. For $n \times n$ matrix polynomials whose leading coefficient matrix is singular (or even for $m \times n$ matrix polynomials), linearizations of smaller size than the ones given in Definition 1.3.1 have been considered in [32] and sharp lower bounds on their sizes have been provided in [41].

1.4 Jordan chains and root polynomials

In classical linear algebra, the concept of Jordan chains for constant matrices is relatively well-known in connection with the eigenvalue problem (see [45, 69]). The extension of this notion to matrix polynomials is defined as follows.

Definition 1.4.1. *Let $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ be a regular matrix polynomial and consider an eigenvalue $\lambda_0 \in \sigma(L)$.*

1. *A sequence of vectors $v_0 \neq 0, v_1, \ldots, v_{k-1}$ ($k \in \mathbb{N}^*$) of $\overline{\mathbb{K}}^n$ is called a* Jordan chain of length *k for $L(\lambda)$ associated with λ_0 if these vectors satisfy*

$$\begin{cases} L(\lambda_0)\, v_0 = 0, \\ \frac{1}{1!}L'(\lambda_0)\, v_0 + L(\lambda_0)\, v_1 = 0, \\ \qquad\qquad \vdots \\ \frac{1}{(k-1)!}L^{(k-1)}(\lambda_0)\, v_0 + \cdots + \frac{1}{1!}L'(\lambda_0)\, v_{k-2} + L(\lambda_0)\, v_{k-1} = 0, \end{cases}$$

   where $L^{(i)}(\lambda)$ represents the ith derivative with respect to λ of the matrix polynomial $L(\lambda)$.

2. *If $v_0, v_1, \ldots, v_{k-1}$ ($k \in \mathbb{N}^*$) is a Jordan chain for $L(\lambda)$ of length k associated with λ_0, then, by definition, v_0 is an eigenvector of $L(\lambda)$ associated with λ_0. The other vectors $v_1, \ldots, v_{k-1}$ are called* generalized eigenvectors *associated with λ_0.*

Remark that a Jordan chain for a matrix $A \in \mathbb{K}^{n\times n}$ associated with an eigenvalue λ_0 (see [45]) is a Jordan chain for the matrix pencil $I_n\lambda - A$ (in the sense of Definition 1.4.1) associated with the same eigenvalue. In this case, it has been shown that the vectors in a Jordan chain are linearly independent. Nevertheless this is not always true for matrix polynomials of degree greater than one where the generalized eigenvectors can be zero.

Definition 1.4.2. *Let $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ be a regular matrix polynomial and consider an eigenvalue $\lambda_0 \in \sigma(L)$. A vector polynomial $\phi(\lambda) \in \overline{\mathbb{K}}[\lambda]^n$ is called a* root polynomial *of $L(\lambda)$ of* order *$k \in \mathbb{N}^*$ associated with λ_0 if $\phi(\lambda_0) \neq 0$ and $L(\lambda)\phi(\lambda) = 0 \mod (\lambda - \lambda_0)^k$.*

From Definitions 1.4.1 and 1.4.2, it follows that, for a given eigenvalue λ_0 of $L(\lambda)$, there exists a correspondence between the set of root polynomials and the set of Jordan chains. Indeed, if one writes a root polynomial $\phi(\lambda)$ of $L(\lambda)$ of order k associated with λ_0 as a polynomial in $\lambda - \lambda_0$

$$\phi(\lambda) = \sum_{i=0}^{q} (\lambda - \lambda_0)^i\, \phi_i,$$

where $q \in \mathbb{N}$, $\phi_i \in \overline{\mathbb{K}}^n$ and $\phi_0, \phi_q \neq 0$, then, for any positive integer $p \leq k$, the vectors $\phi_0, \ldots, \phi_{p-1}$, with the convention $\phi_i = 0$ when $i > q$, form a Jordan chain of length p associated with λ_0. This can be seen as follows. Taylor's formula applied to $L(\lambda)$ at the point λ_0 gives

$$L(\lambda) = \sum_{j=0}^{\ell} \frac{1}{j!} L^{(j)}(\lambda_0)\, (\lambda - \lambda_0)^j\,.$$

The product $L(\lambda)\phi(\lambda)$ is thus equal to

$$L(\lambda)\phi(\lambda) = \left[\sum_{j=0}^{\ell} \frac{1}{j!} L^{(j)}(\lambda_0)\,(\lambda-\lambda_0)^j\right]\left[\sum_{i=0}^{q}(\lambda-\lambda_0)^i\,\phi_i\right]$$

$$= \sum_{i=0}^{p-1}\left[\sum_{j=0}^{i}\frac{1}{j!}L^{(j)}(\lambda_0)\,\phi_{i-j}\right](\lambda-\lambda_0)^i \mod (\lambda-\lambda_0)^p\,. \tag{1.9}$$

Since $L(\lambda)\phi(\lambda) = 0 \mod (\lambda-\lambda_0)^k$ with $k \geq p$, it follows from relation (1.9) that for $i = 0,\ldots,p-1$, one has

$$\sum_{j=0}^{i}\frac{1}{j!}L^{(j)}(\lambda_0)\,\phi_{i-j} = 0. \tag{1.10}$$

Therefore, $\phi_0,\ldots,\phi_{p-1}$ form a Jordan chain of length p associated with λ_0. Conversely, if $v_0, v_1, \ldots, v_{p-1}$ is a Jordan chain of length p for $L(\lambda)$ associated with λ_0, then the vector polynomial defined by

$$\phi(\lambda) = \sum_{i=0}^{p-1}(\lambda-\lambda_0)^i\,v_i + (\lambda-\lambda_0)^p\,\psi(\lambda),$$

where $\psi(\lambda)$ is an arbitrary vector polynomial of $\overline{\mathbb{K}}[\lambda]^n$, is a root polynomial of $L(\lambda)$ of order p associated with λ_0.

Definition 1.4.3. *Let $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ be a matrix polynomial and $\lambda_0 \in \sigma(L)$. An eigenvector v_0 associated with λ_0 is said to be of* rank κ *if the maximal order of root polynomials $\phi(\lambda)$ of $L(\lambda)$ associated with λ_0 such that $\phi(\lambda_0) = v_0$ is κ.*

Due to the relation between Jordan chains and root polynomials, we can say that an eigenvector v_0 of $L(\lambda)$ associated with λ_0 is of rank κ if the maximum length of Jordan chains for $L(\lambda)$ associated with λ_0 having v_0 as a first vector is κ.

Theorem 1.4.1 ([50], Th. 1.12)**.** *Let $L(\lambda)$ be a regular matrix polynomial and $\lambda_0 \in \sigma(L)$. The rank of an eigenvector v_0 of $L(\lambda)$ associated with λ_0 is always equal to one of the partial multiplicities of $L(\lambda)$ at λ_0.*

We illustrate the notions of Jordan chain and rank of an eigenvector with the following example.

Example 1.4.1. *Let*

$$L(\lambda) = \begin{pmatrix} \lambda^2 - 2\,\lambda + 1 & -2\,\lambda^5 + 2\,\lambda^4 + 3\,\lambda^3 - 4\,\lambda^2 + \lambda \\ -2\,\lambda^3 + 5\,\lambda^2 - 4\,\lambda + 1 & 2\,\lambda^6 - 8\,\lambda^4 + 7\,\lambda^3 - \lambda \end{pmatrix}$$

be the matrix polynomial given by (1.2). We have found in Example 1.1.2 that any nonzero vector of $\mathbb{C}^2$ is an eigenvector of $L(\lambda)$ associated with the eigenvalue 1. Let us now compute the Jordan chains for $L(\lambda)$ associated with 1 which begin with an eigenvector $v_0 = \begin{pmatrix} v_{01} \\ v_{02} \end{pmatrix}$. For the first generalized eigenvector $v_1 = \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix}$, we have

$$L'(1)\,v_0 + L(1)\,v_1 = 0 \iff \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{01} \\ v_{02} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = 0.$$

This latter relation holds for any vector v_1 *of* $\mathbb{C}^2$ *and any (nonzero) vector* v_0 *of* $\mathbb{C}^2$*. For the second generalized eigenvector* $v_2 = \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$*, we have*

$$\begin{aligned} & \tfrac{1}{2!}L^{(2)}(1)\,v_0 + L'(1)\,v_1 + L(1)\,v_2 = 0 \\ \iff \quad & \begin{pmatrix} 1 & -3 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} v_{01} \\ v_{02} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = 0 \\ \implies \quad & v_{01} = 3\,v_{02}. \end{aligned}$$

Thus, v_2 *exists if and only if* v_0 *satisfies* $v_{01} = 3\,v_{02}$*. Consequently, any Jordan chain associated with eigenvalue* 1 *beginning with an eigenvector of the form* $v_0 = \begin{pmatrix} v_{01} \\ v_{02} \end{pmatrix}$ *such that* $v_{01} \neq 3\,v_{02}$ *is of length at most* 2*. Assume now that the eigenvector* v_0 *satisfies* $v_{01} = 3\,v_{02}$*. For the third generalized eigenvector* $v_3 = \begin{pmatrix} v_{31} \\ v_{32} \end{pmatrix}$*, we have*

$$\begin{aligned} & \tfrac{1}{3!}L^{(3)}(1)\,v_0 + \tfrac{1}{2!}L^{(2)}(1)\,v_1 + L'(1)\,v_2 + L(1)\,v_3 = 0 \\ \iff \quad & \begin{pmatrix} 0 & -9 \\ -2 & 15 \end{pmatrix}\begin{pmatrix} 3\,v_{02} \\ v_{02} \end{pmatrix} + \begin{pmatrix} 1 & -3 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_{31} \\ v_{32} \end{pmatrix} = 0 \\ \implies \quad & -9\,v_{02} + v_{11} - 3\,v_{12} = 0 \\ \implies \quad & v_{11} = 9\,v_{02} + 3\,v_{12}. \end{aligned}$$

Thus, the first generalized eigenvector v_1 *of any Jordan chain associated with the eigenvalue* 1 *of length at least* 4 *must satisfy* $v_{11} = 9\,v_{02} + 3\,v_{12}$*. For the fourth generalized eigenvector* v_4*, we have*

$$\begin{aligned} & \tfrac{1}{4!}L^{(4)}(1)\,v_0 + \tfrac{1}{3!}L^{(3)}(1)\,v_1 + \tfrac{1}{2!}L^{(2)}(1)\,v_2 + L'(1)\,v_3 + L(1)\,v_4 = 0 \\ \iff \quad & \begin{pmatrix} 0 & -8 \\ 0 & 22 \end{pmatrix}\begin{pmatrix} 3\,v_{02} \\ v_{02} \end{pmatrix} + \begin{pmatrix} 0 & -9 \\ -2 & 15 \end{pmatrix}\begin{pmatrix} 9\,v_{02} + 3\,v_{12} \\ v_{12} \end{pmatrix} + \begin{pmatrix} 1 & -3 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = 0 \\ \implies \quad & \begin{cases} -8\,v_{02} - 9\,v_{12} + v_{21} - 3\,v_{22} = 0 \\ 4\,v_{02} + 9\,v_{12} - v_{21} + 3\,v_{22} = 0 \end{cases} \\ \implies \quad & v_{02} = 0. \end{aligned}$$

This implies $v_0 = 0$ *which is impossible. Thus, the length of any Jordan chain associated with* 1 *beginning with an eigenvector of the form* $v_0 = \begin{pmatrix} 3\,v_{02} \\ v_{02} \end{pmatrix}$ $(v_{02} \neq 0)$ *is at most* 4*. To summarize, the Jordan chains associated with eigenvalue* 1 *can be described as follows:*

- *Jordan chains of length* 1 *are formed by an arbitrary nonzero vector* v_0 *of* $\mathbb{C}^2$*;*
- *Jordan chains of length* 2 *are of the form* v_0, v_1*, where* v_0, v_1 *are both arbitrary vectors of* $\mathbb{C}^2$ *and* $v_0 \neq 0$*;*
- *Jordan chains of length* 3 *are of the form* $v_0 = \begin{pmatrix} 3\,v_{02} \\ v_{02} \end{pmatrix}, v_1, v_2$*, where* $v_{02} \in \mathbb{C} \setminus \{0\}$ *and* v_1 *and* v_2 *are two arbitrary vectors of* $\mathbb{C}^2$*;*
- *Jordan chains of length* 4 *are of the form* $v_0 = \begin{pmatrix} 3\,v_{02} \\ v_{02} \end{pmatrix}, v_1 = \begin{pmatrix} 9\,v_{02} + 3\,v_{12} \\ v_{12} \end{pmatrix}, v_2, v_3$*, where* $v_{02} \in \mathbb{C} \setminus \{0\}$*,* $v_{12} \in \mathbb{C}$ *and* v_2 *and* v_3 *are two arbitrary vectors of* $\mathbb{C}^2$*;*

From above explanation, we can deduce that the rank of an eigenvector $v_0 = \begin{pmatrix} v_{01} \\ v_{02} \end{pmatrix}$ *associated with the eigenvalue* 1 *is equal to* 2 *if* $v_{01} \neq 3\,v_{02}$ *and* 4 *otherwise. We recall that* 2 *and* 4 *are indeed the partial multiplicities of* $L(\lambda)$ *at* 1 *(see Example 1.2.2).* ■

1.5 Canonical sets of Jordan chains

In Theorem 1.4.1, we have seen that the rank of an eigenvector associated with a given eigenvalue λ_0 is equal to one of the partial multiplicities at λ_0. The following proposition states that it is always possible to find a basis of the right nullspace of $L(\lambda_0)$ (*i.e.*, $m_g(\lambda_0)$ linearly independent eigenvectors associated with λ_0) such that each partial multiplicity of $L(\lambda)$ at λ_0 corresponds to the rank of an element of this basis.

Proposition 1.5.1 ([50], page 32)**.** *Let* $L(\lambda)$ *be a regular matrix polynomial and* $\lambda_0 \in \sigma(L)$ *an eigenvalue with partial multiplicities* $\kappa_1, \ldots, \kappa_{m_g(\lambda_0)}$*. For each* $i = 1, \ldots, m_g(\lambda_0)$*, there exists a Jordan chain* $v_{i,0}, \ldots, v_{i,\kappa_i - 1}$ *of maximal length* κ_i *associated with* λ_0*,* i.e.*, there exists an eigenvector* $v_{i,0}$ *of rank* κ_i*, such that the family of eigenvectors* $\left(v_{1,0}, \ldots, v_{m_g(\lambda_0),0}\right)$ *forms a basis of the right nullspace of* $L(\lambda_0)$*.*

Definition 1.5.1. *With the notation of Proposition 1.5.1, the Jordan chains*

$$v_{1,0}, \ldots, v_{1,\kappa_1 - 1}, \quad v_{2,0}, \ldots, v_{2,\kappa_2 - 1}, \quad \ldots, \quad v_{m_g(\lambda_0),0}, \ldots, v_{m_g(\lambda_0),\kappa_{m_g(\lambda_0)} - 1}$$

are said to form a canonical set *of Jordan chains for* $L(\lambda)$ *associated with* λ_0*.*

We draw attention that a canonical set of Jordan chains is not unique. Moreover, not every basis of the right nullspace of $L(\lambda_0)$ can be extended to form a canonical set of Jordan chains at λ_0.

Example 1.5.1. *We consider the matrix polynomial* $L(\lambda)$ *defined by (1.2). From the calculations done in Example 1.4.1, we can deduce that the Jordan chains*

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

form a canonical set of Jordan chains for $L(\lambda)$ *associated with the eigenvalue* 1*. Another canonical set of Jordan chains associated with* 1 *could be*

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ i \end{pmatrix}, \qquad \begin{pmatrix} 3\,i \\ i \end{pmatrix}, \begin{pmatrix} 9\,i + 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}.$$

Consider now the basis of $\ker(L(\lambda_0))$ *composed of the eigenvectors* $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ *and* $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$*. This basis cannot be extended to form a canonical set of Jordan chains associated with* 1 *since these two eigenvectors have rank equal to* 2*.* ■

Proposition 1.5.2 ([50], Prop. 1.15)**.** *Given a regular matrix polynomial* $L(\lambda)$ *and an eigenvalue* λ_0 *with partial multiplicities* $\kappa_1, \ldots, \kappa_{m_g(\lambda_0)}$*, a set of Jordan chains for* $L(\lambda)$

$$v_{1,0}, \ldots, v_{1,\kappa_1 - 1}, \quad v_{2,0}, \ldots, v_{2,\kappa_2 - 1}, \quad \ldots, \quad v_{m_g(\lambda_0),0}, \ldots, v_{m_g(\lambda_0),\kappa_{m_g(\lambda_0)} - 1}$$

associated with λ_0 *form a canonical set if and only if the eigenvectors* $v_{1,0}, \ldots, v_{m_g(\lambda_0),0}$ *are linearly independent.*

1.6 Efficient computation of a canonical set of Jordan chains

The partial multiplicities at an eigenvalue λ_0 and the corresponding Jordan chains in a canonical set can be computed respectively from the local Smith form at λ_0 and from the inverse of the multiplier $F_{\lambda_0}(\lambda)$ (see Theorem 1.2.2) which can be supposed to be a unimodular matrix polynomial (as we did in Example 1.2.2 by absorbing the missing part of the Smith form $S(\lambda)$ into the multiplier on the left $E(\lambda)$). In this section, we investigate more efficient methods for obtaining a canonical set of Jordan chains for an $n \times n$ regular matrix polynomial $L(\lambda)$ at an eigenvalue λ_0 avoiding the computation of the (local) Smith form of $L(\lambda)$.

If we assume that we know one partial multiplicity κ of $L(\lambda)$ at λ_0, then, according to Definition 1.4.1, the corresponding Jordan chains in a canonical set are among the solutions of the linear system

$$T_{\lambda_0,i}\mathcal{V}_i = \begin{pmatrix} L(\lambda_0) & 0 & \cdots & \cdots & 0 \\ \frac{1}{1!}L'(\lambda_0) & L(\lambda_0) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{1}{(i-2)!}L^{(i-2)}(\lambda_0) & \cdots & \ddots & L(\lambda_0) & 0 \\ \frac{1}{(i-1)!}L^{(i-1)}(\lambda_0) & \cdots & \cdots & \frac{1}{1!}L'(\lambda_0) & L(\lambda_0) \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{i-2} \\ v_{i-1} \end{pmatrix} = 0, \quad \text{for } i = \kappa. \tag{1.11}$$

For the computation of the partial multiplicities at λ_0, it has been shown in [104, Chap. 3] (see also [97]) that they can be obtained by analyzing iteratively the ranks of the *block Toeplitz matrices* $T_{\lambda_0,i}$ given in (1.11) as stated in the following proposition.

Proposition 1.6.1 ([104], Prop. 3.1)**.** *Let $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ be a regular matrix polynomial of degree ℓ, $\lambda_0 \in \sigma(L)$ and for $i \geq 1$, $T_{\lambda_0,i}$ the block Toeplitz matrix appearing in (1.11). Let r_1 denote the rank of $T_{\lambda_0,1} = L(\lambda_0)$ and let, for $i \geq 2$, $r_i = \operatorname{rank}(T_{\lambda_0,i}) - \operatorname{rank}(T_{\lambda_0,i-1})$. Then*

1. *$\forall\, i \in \mathbb{N}^*$, $r_i \leq r_{i+1}$;*
2. *if the difference $x_i = r_{i+1} - r_i$ is a positive integer, then there exists x_i partial multiplicities at λ_0 equal to i;*
3. *there exists a positive integer $p \leq m_a(\lambda_0) + 1 \leq n\,\ell + 1$ such that $\forall\, i \geq p$, $r_i = n$ and hence one has $\sum_{i=1}^{p-1} x_i = m_g(\lambda_0)$.*

Example 1.6.1. *Consider the 2×2 matrix polynomial $L(\lambda)$ given by (1.2) and its eigenvalue $\lambda_0 = 1$. Let us compute the integers r_i defined in Proposition 1.6.1. For this, we construct successively the block Toeplitz matrices $T_{1,i}$ for $i \geq 1$ then compute their ranks and the integers r_i. We find successively $r_1 = 0$, $r_2 = 0$, $r_3 = 1$, $r_4 = 1$ and $r_5 = 2$ the step at which we stop since we find an integer p ($p = 5$) for which $r_p = n = 2$. Thus, the differences $x_i = r_{i+1} - r_i$ for $i = 1, \ldots, p-1$, are equal to $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ and $x_4 = 1$. This means that there exist two partial multiplicities at $\lambda_0 = 1$, one equal to 2 and the other equal to 4. This is consistent with what we found in Example 1.2.2.* ■

Thus, based on Proposition 1.6.1, one can deduce a first algorithm for computing a canonical set of Jordan chains at λ_0. This algorithm proceeds as follows. One constructs successively the block Toeplitz matrices $T_{\lambda_0,i}$ for $i \geq 1$, and computes the integers r_i and x_i as defined in Proposition 1.6.1. If for some $i = \kappa$, $x_\kappa \neq 0$, then one computes a basis of solutions of System (1.11) for $i = \kappa$ and stores its elements in a set $\mathcal{E}_\kappa$. One continues this process until finding a positive integer p for which $r_p = n$. At this stage, one knows that the set of partial multiplicities at λ_0 has been completed. We point out that each element of the set $\mathcal{E}_\kappa$ corresponds to a Jordan chain associated with λ_0 of length less than or equal to κ. Indeed, a sequence of vectors

$v_0, v_1, \ldots, v_{j-1}$ of $\overline{\mathbb{K}}^n$ form a Jordan chain for $L(\lambda)$ of length $j \leq \kappa$ associated with λ_0 if and only if the vector

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ v_0 \\ \vdots \\ v_{j-1} \end{pmatrix} \in \overline{\mathbb{K}}^{n\kappa}$$

is a solution of System (1.11) for $i = \kappa$. Therefore, to obtain a canonical set of Jordan chains for $L(\lambda)$ at λ_0, the last step of the algorithm consists in picking respectively x_κ elements from $\mathcal{E}_\kappa$ corresponding to Jordan chains of length equal to κ, such that the eigenvectors of all Jordan chains collected from all sets $\mathcal{E}_\kappa$ are linearly independent. This can be done as follows. Let $\kappa_1 < \cdots < \kappa_s$ be the distinct partial multiplicities at λ_0. Select x_{κ_s} elements from $\mathcal{E}_{\kappa_s}$ corresponding to Jordan chains of length κ_s and whose eigenvectors are linearly independent. Put them in a set $\mathcal{X}_{\kappa_s}$. Then, choose $x_{\kappa_{s-1}}$ elements from $\mathcal{E}_{\kappa_{s-1}}$ corresponding to Jordan chains of length κ_{s-1} and whose eigenvectors are linearly independent and not in the span of the eigenvectors of the Jordan chains in $\mathcal{X}_{\kappa_s}$. Put these elements in a set $\mathcal{X}_{\kappa_{s-1}}$, and so on. If we suppose that $\mathcal{X}_{\kappa_s}, \ldots, \mathcal{X}_{\kappa_{i+1}}$ have been already determined, then choose x_{κ_i} elements from $\mathcal{E}_{\kappa_i}$ corresponding to Jordan chains of length κ_i and whose eigenvectors are linearly independent and not in the span of the eigenvectors of the Jordan chains in $\mathcal{X}_{\kappa_s}, \ldots, \mathcal{X}_{\kappa_{i+1}}$. We continue this process until $\mathcal{X}_{\kappa_1}$ is determined. In this way, the elements of $\mathcal{X}_{\kappa_1}, \ldots, \mathcal{X}_{\kappa_s}$ form a canonical set of Jordan chains for $L(\lambda)$ at λ_0 (see Proposition 1.5.2 above).

In the above approach, computing directly the rank, respectively the right nullspace, of matrix $T_{\lambda_0,i}$, which is of size $i\,n \times i\,n$, without taking into consideration its particular block structure could be expensive. Indeed, if we suppose that computing the rank, respectively the right nullspace, of an $n \times n$ matrix with entries in $\mathbb{K}(\lambda_0)$ costs $O(n^\omega\, d_{\lambda_0})$ operations in $\mathbb{K}$, where d_{λ_0} denotes the degree of the extension $\mathbb{K}(\lambda_0)$ over $\mathbb{K}$, then computing the rank, respectively the right nullspace, of $T_{\lambda_0,i}$ costs at most $O(n^\omega\, i^\omega\, d_{\lambda_0})$ operations in $\mathbb{K}$ with $i \leq m_a(\lambda_0) \leq n\,\ell$. The latter inequalities follow from the fact that the values of integers i in Proposition 1.6.1 give those of the partial multiplicities at λ_0 which are bounded by $m_a(\lambda_0) \leq n\,\ell$. Thus, it is worth looking for other efficient ways for computing the integers r_i and the right nullspaces of matrices $T_{\lambda_0,i}$.

It has been noticed in [104, Chap. 3] that the integers r_i can be obtained by analyzing only the rank of the last block row of matrices $T_{\lambda_0,i}$. Indeed, for $i \geq 2$, let $T^*_{\lambda_0,i-1}$ denote the submatrix of $T_{\lambda_0,i}$ composed of rows of indices from $(i-1)\,n+1$ to $i\,n$ and columns of indices from 1 to $(i-1)\,n$ respectively, namely,

$$T^*_{\lambda_0,i-1} = \begin{pmatrix} \frac{1}{(i-1)!}L^{(i-1)}(\lambda_0) & \cdots & L'(\lambda_0) \end{pmatrix} \in \overline{\mathbb{K}}^{n\times(i-1)n}.$$

Thus, for $i \geq 2$, matrix $T_{\lambda_0,i}$ can be written as

$$T_{\lambda_0,i} = \begin{pmatrix} T_{\lambda_0,i-1} & 0 \\ T^*_{\lambda_0,i-1} & L(\lambda_0) \end{pmatrix}. \tag{1.12}$$

Remark now that the integer r_i defined by $r_i = \operatorname{rank}(T_{\lambda_0,i}) - \operatorname{rank}(T_{\lambda_0,i-1})$ is the number of linearly independent rows added in the block row $\begin{pmatrix} T^*_{\lambda_0,i-1} & L(\lambda_0) \end{pmatrix}$. Let V be a matrix whose columns form a basis of $\ker(T_{\lambda_0,i-1})$. Augment matrix V by another matrix V^* in such a way that the columns of the matrix $\begin{pmatrix} V^* & V \end{pmatrix}$ form a basis of the $\mathbb{K}(\lambda_0)$-vector space $\mathbb{K}(\lambda_0)^{(i-1)\,n}$. Therefore, one has

$$\operatorname{rank}(T_{\lambda_0,i}) = \operatorname{rank}\left(T_{\lambda_0,i} \begin{pmatrix} V^* & V & 0 \\ 0 & 0 & I_n \end{pmatrix} \right)$$

with, from (1.12),

$$T_{\lambda_0,i}\begin{pmatrix} V^* & V & 0 \\ 0 & 0 & I_n \end{pmatrix} = \begin{pmatrix} T_{\lambda_0,i-1}V^* & T_{\lambda_0,i-1}V & 0 \\ T^*_{\lambda_0,i-1}V^* & T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix} = \begin{pmatrix} T_{\lambda_0,i-1}V^* & 0 & 0 \\ T^*_{\lambda_0,i-1}V^* & T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix}.$$

Since matrix $T_{\lambda_0,i-1}V^*$ is of full column-rank and $\operatorname{rank}(T_{\lambda_0,i-1}V^*) = \operatorname{rank}(T_{\lambda_0,i-1})$, it follows that

$$\begin{aligned} r_i &= \operatorname{rank}(T_{\lambda_0,i}) - \operatorname{rank}(T_{\lambda_0,i-1}) \\ &= \operatorname{rank}\left(T_{\lambda_0,i}\begin{pmatrix} V^* & V & 0 \\ 0 & 0 & I_n \end{pmatrix}\right) - \operatorname{rank}(T_{\lambda_0,i-1}V^*) \\ &= \operatorname{rank}\begin{pmatrix} T_{\lambda_0,i-1}V^* & 0 & 0 \\ T^*_{\lambda_0,i-1}V^* & T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix} - \operatorname{rank}(T_{\lambda_0,i-1}V^*) \\ &= \operatorname{rank}\begin{pmatrix} T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix}. \end{aligned} \tag{1.13}$$

Lemma 1.6.1. *Given a regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ and an eigenvalue $\lambda_0 \in \sigma(L)$, the dimension of the right nullspace of matrix $T_{\lambda_0,i}$ $(i \geq 1)$ defined in (1.11) is less than or equal to $m_a(\lambda_0)$.*

Proof. From Proposition 1.6.1, we can deduce that $\operatorname{rank}(T_{\lambda_0,i}) = \sum_{j=1}^{i} r_j = i\,(n - m_g(\lambda_0)) + \sum_{j=1}^{i-1}(i-j)\,x_j$. Thus,

$$\begin{aligned} \dim(\ker(T_{\lambda_0,i})) &= i\,n - \operatorname{rank}(T_{\lambda_0,i}) \\ &= i\,m_g(\lambda_0) - \sum_{j=1}^{i-1}(i-j)\,x_j \\ &= i\left(m_g(\lambda_0) - \sum_{j=1}^{i-1} x_j\right) + \sum_{j=1}^{i-1} j\,x_j \\ &= i\sum_{j=i}^{p-1} x_j + \sum_{j=1}^{i-1} j\,x_j \\ &\leq \sum_{j=1}^{p-1} j\,x_j, \end{aligned}$$

where p is the integer given in Proposition 1.6.1. Now $\sum_{j=1}^{p-1} j\,x_j$ is the sum of the partial multiplicities at λ_0 which is equal to $m_a(\lambda_0)$. □

Thus, from relation (1.13) and Lemma 1.6.1, it follows that the integer r_i is the rank of a rectangular matrix of size $n \times m$ with $m \leq m_a(\lambda_0) + n$. It is known that computing the rank of a rectangular matrix of size $n \times m$ with coefficients in $\mathbb{K}(\lambda_0)$ can be done in $O\big(n^2\,m\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$ (using, *e.g.*, Gaussian elimination). Hence, computing integer r_i can be done in at most $O\big(n^2\,(m_a(\lambda_0)+n)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$. In this way, one has reduced the number of operations required at each step. Now, it remains to explain how one can obtain a basis of the right nullspace of matrix $T_{\lambda_0,i}$ from those of $T_{\lambda_0,i-1}$ and $\begin{pmatrix} T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix}$. Let W denote a matrix whose columns form a basis of the right nullspace of the matrix $\begin{pmatrix} T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix}$. The columns of the matrix

$$V' = \begin{pmatrix} V & 0 \\ 0 & I_n \end{pmatrix} W$$

form a basis of $\ker(T_{\lambda_0,i})$. Indeed, it is easy to check that

$$T_{\lambda_0,i}V' = \begin{pmatrix} T_{\lambda_0,i-1} & 0 \\ T^*_{\lambda_0,i-1} & L(\lambda_0) \end{pmatrix} \begin{pmatrix} V & 0 \\ 0 & I_n \end{pmatrix} W = 0,$$

which means that the columns of matrix V' belong to $\ker(T_{\lambda_0,i})$. Furthermore, these columns are linearly independent since V' is the product of two matrices of full-column rank. Finally, the number of columns of V' is equal to

$$\begin{aligned} \operatorname{coldim}(V') &= \operatorname{coldim}(W) \\ &= \operatorname{coldim}\begin{pmatrix} T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix} - \operatorname{rank}\begin{pmatrix} T^*_{\lambda_0,i-1}V & L(\lambda_0) \end{pmatrix} \\ &= \operatorname{coldim}(V) + n - r_i \\ &= (i-1)\,n - \operatorname{rank}(T_{\lambda_0,i-1}) + n - r_i \\ &= i\,n - \operatorname{rank}(T_{\lambda_0,i-1}) - r_i \\ &= i\,n - \operatorname{rank}(T_{\lambda_0,i}) \\ &= \dim(\ker(T_{\lambda_0,i}))\,, \end{aligned}$$

hence the columns of matrix V' form a basis of $\ker(T_{\lambda_0,i})$.

This provides another algorithm[1] for computing a canonical set of Jordan chains at a given eigenvalue λ_0. Algorithm **CanonicalSet_JC** below takes as input a regular matrix polynomial $L(\lambda)$ and an eigenvalue $\lambda_0 \in \sigma(L)$ and returns the partial multiplicities of $L(\lambda)$ at λ_0 with a canonical set of Jordan chains for $L(\lambda)$ at λ_0. More precisely, the output is a list of elements of the form

$$[\kappa, x_\kappa, \mathrm{JC}_\kappa\,],$$

where κ is a partial multiplicity of $L(\lambda)$ at λ_0, x_κ is the number of partial multiplicities equal to κ (*i.e.*, κ occurs x_κ times in the local Smith form at λ_0) and JC_κ is a list of x_κ vectors of size $n\,\kappa$ of the form

$$\begin{pmatrix} v_0 \\ \vdots \\ v_{\kappa-1} \end{pmatrix}.$$

The latter vector corresponds to a Jordan chain $v_0, \ldots, v_{\kappa-1}$ at λ_0 such that the rank of the eigenvector v_0 is equal to κ. Thus, the Jordan chains corresponding to the elements of all lists JC_κ form a canonical set of Jordan chains for $L(\lambda)$ at λ_0.

ALGORITHM **CanonicalSet_JC**

INPUT: A regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ and an eigenvalue $\lambda_0 \in \sigma(L)$.
OUTPUT: The partial multiplicities of $L(\lambda)$ at λ_0 and a canonical set of Jordan chains for $L(\lambda)$ at λ_0.

INITIALIZATION: Put $T_1 = L(\lambda_0)$, $i = 2$ and PM = { };

1. Compute a matrix V_1 whose columns form a basis of $\ker(T_1)$;
   Let $r_1 = n - \operatorname{coldim}(V_1)$; // r_1 is the rank of T_1
2. **While** $r_{i-1} \neq n$ **do**
   2.1. Compute $\overline{A}_{i-1} = \frac{1}{(i-1)!}L^{(i-1)}(\lambda_0)$ and put $T^* = \begin{pmatrix} \overline{A}_{i-1} & \overline{A}_{i-2} & \cdots & \overline{A}_1 \end{pmatrix}$;
   2.2. Compute T^*V_{i-1} and set $T = \begin{pmatrix} T^*V_{i-1} & T_1 \end{pmatrix}$;

[1] A slightly modified version of this algorithm has been presented in [104, Algorithm 3.3]. Another algorithm is given in [102].

2.3. Compute a matrix W whose columns form a basis of $\ker(T)$;
Let $r_i = \text{coldim}(T) - \text{coldim}(W)$; // r_i is the rank of T
2.4. Let $x_{i-1} = r_i - r_{i-1}$;
If $x_{i-1} \neq 0$ **then**
Let PM= PM $\cup \{i-1\}$;
end if;
2.5. Compute $V_i = \text{diag}(V_{i-1}, I_n)\, W$;
// The columns of V_i form a basis of the right nullspace of $T_{\lambda_0,i}$ defined in (1.11)
2.6. Put $i = i + 1$;
end do;
3. For each $\kappa \in$ PM, extract "carefully" x_κ columns of V_κ and put them in a list JC_κ;
// See Remark 1.6.1 below
4. **Return** $[[\kappa, x_\kappa, \text{JC}_\kappa]$, for $\kappa \in \text{PM}]$;

Remark 1.6.1. In step 3 of Algorithm **CanonicalSet_JC**, one has to extract x_κ columns from matrix V_κ in such a way that they correspond to Jordan chains of length κ and that the eigenvectors of these Jordan chains for all values $\kappa \in$ PM are linearly independent. To achieve this, one can proceed as we have explained earlier. In our implementation, we proceed differently. For each $\kappa \in$ PM, we take x_κ random linear combinations of the columns of V_κ corresponding to Jordan chains of length κ. Then, we check whether the eigenvectors of these linear combinations for all $\kappa \in$ PM are linearly independent. If this is not the case, we iterate the process.

We have implemented[2] Algorithm **CanonicalSet_JC** in Maple. We illustrate it on the matrix polynomial given by (1.2).

Example 1.6.2. *We first enter the matrix polynomial given by (1.2) in* Maple

```
> L := Matrix([[x^2-2*x+1,-2*x^5+2*x^4+3*x^3-4*x^2+x],[5*x^2-4*x+1-2*x^3,
> 2*x^6-8*x^4+7*x^3-x]]);
```

$$L := \begin{bmatrix} x^2 - 2\,x + 1 & -2\,x^5 + 2\,x^4 + 3\,x^3 - 4\,x^2 + x \\ 5\,x^2 - 4\,x + 1 - 2\,x^3 & 2\,x^6 - 8\,x^4 + 7\,x^3 - x \end{bmatrix}$$

Now, we apply the procedure **CanonicalSet_JC** *to L at the eigenvalue* 1

```
> CanonicalSet_JC(L,x,1);
```

$$\left[\left[2, 1, \left[\left[\begin{bmatrix} 10 \\ 6 \\ 0 \\ 0 \end{bmatrix}\right]\right]\right], \left[4, 1, \left[\left[\begin{bmatrix} -\frac{2}{3} \\ -\frac{2}{9} \\ 4 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right]\right]\right]\right]$$

[2]The code is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`.

This means that the two chains

$$\begin{pmatrix}10\\6\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix} \quad \textit{and} \quad \begin{pmatrix}-\frac{2}{3}\\-\frac{2}{9}\end{pmatrix}, \begin{pmatrix}4\\2\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}$$

form a canonical set of Jordan chains for $L(\lambda)$ associated with eigenvalue 1. ■

Proposition 1.6.2. *Given a regular matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ of degree ℓ and an eigenvalue $\lambda_0 \in \sigma(L)$, Algorithm* **CanonicalSet_JC** *computes a canonical set of Jordan chains for $L(\lambda)$ at λ_0 in at most $O\big(\big(n\, m_a(\lambda_0)^2 \min\{m_a(\lambda_0), \ell\} + n^2\, m_a(\lambda_0) + m_a(\lambda_0)^4 + n\,\ell^2\big)\, n\, d_{\lambda_0}\big)$ arithmetic operations in $\mathbb{K}$, where d_{λ_0} denotes the degree of the extension $\mathbb{K}(\lambda_0)$ over $\mathbb{K}$.*

Proof. Let us first investigate the cost of computing $L^{(i)}(\lambda_0)$ for $i = 0, \ldots, \ell$. For this, we need to differentiate n^2 polynomials of degree bounded by ℓ at most ℓ times. Each differentiation costs $O(\ell)$ arithmetic operations in $\mathbb{K}$. Thus, the cost of computing the derivatives $L^{(i)}(\lambda)$ for $i = 0, \ldots, \ell$ is bounded by $O\big(n^2\,\ell^2\big)$ operations in $\mathbb{K}$. Then, we need to evaluate each of these derivatives at λ_0. This is equivalent to evaluate at most $n^2\,\ell$ polynomials of degree bounded by ℓ at λ_0. Since evaluating a polynomial of degree $\leq \ell$ with coefficients in $\mathbb{K}$ at λ_0 can be done using at most $O(\ell\, d_{\lambda_0})$ operations in $\mathbb{K}$, where d_{λ_0} denotes the degree of the extension $\mathbb{K}(\lambda_0)$ over $\mathbb{K}$, the cost of evaluation is $O\big(n^2\,\ell^2\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Hence, computing $L^{(i)}(\lambda_0)$ for $i = 0, \ldots, \ell$ can be done in at most $O\big(n^2\,\ell^2\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$.

Now, matrix $T_1 = L(\lambda_0)$ in Step 1 of the algorithm is of size $n \times n$, therefore computing its right nullspace V_1 requires at most $O\big(n^3\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$.

Let us now compute the cost of one **While** loop in Algorithm **CanonicalSet_JC**. In Step 2.2, matrix T^* is of size $n \times (i-1)\,n$ and matrix V_{i-1} has at most $m_a(\lambda_0)$ columns (see Lemma 1.6.1). Remark that since $\overline{A}_i = 0$ for $i > \ell$, the product T^*V_{i-1} is indeed the sum of $\min\{i-1, \ell\}$ products of an $n \times n$ matrix by a matrix having at most $m_a(\lambda_0)$ columns. So the cost of the multiplication T^*V_{i-1} is bounded by $O\big(n^2\, m_a(\lambda_0) \min\{i-1, \ell\}\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. In Step 2.3, matrix T has n rows and at most $m_a(\lambda_0) + n$ columns. Thus, Step 2.3 costs at most $O\big(n^2\,(m_a(\lambda_0) + n)\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. It remains to study the cost of Step 2.5. Let us partition the matrix W into two blocks $W = \begin{pmatrix} W_1^T & W_2^T \end{pmatrix}^T$ corresponding to the partition of the matrix $\mathrm{diag}(V_{i-1}, I_n)$. The cost of the product $\mathrm{diag}(V_{i-1}, I_n)\, W$ is thus reduced to the cost of multiplication $V_{i-1}W_1$. Since matrix V_{i-1} has $(i-1)\,n$ rows and at most $m_a(\lambda_0)$ columns and matrix W_1 has at most $m_a(\lambda_0)$ columns, the cost of product $V_{i-1}W_1$ is bounded by $O\big(n\, i\, m_a(\lambda_0)^2\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Hence, one passage in the **While** loop costs at most $O\big(\big(n\, m_a(\lambda_0) \min\{i-1, \ell\} + n^2 + i\, m_a(\lambda_0)^2\big)\, n\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. As the integer i is bounded by $m_a(\lambda_0)$ and the **While** loop is repeated at most $m_a(\lambda_0)$ times, Algorithm **CanonicalSet_JC** computes a canonical set of Jordan chains for $L(\lambda)$ corresponding to the eigenvalue λ_0 in at most $O\big(\big(n\, m_a(\lambda_0)^2 \min\{m_a(\lambda_0), \ell\} + n^2\, m_a(\lambda_0) + m_a(\lambda_0)^4 + n\,\ell^2\big)\, n\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. □

Remark 1.6.2. In the complexity result of Proposition 1.6.2 above, bounding $m_a(\lambda_0)^4$ by $n^3\,\ell^3\, m_a(\lambda_0)$, we find that computing a canonical set of Jordan chains for $L(\lambda)$ at λ_0 using Algorithm **CanonicalSet_JC** can be done using at most $O\big(n^4\,\ell^3\, m_a(\lambda_0)\, d_{\lambda_0}\big)$ operations in $\mathbb{K}$. We will use this bound in the complexity analysis of the algorithms developed in the next chapters.

1.7 Minimal bases of singular matrix polynomials

Unlike the previous sections of this chapter, this section merely concerns singular matrix polynomials (see Definition 1.1.1). As it will be useful in our algorithms, we focus on minimal bases of singular matrix polynomials and their computations.

Throughout this section, we suppose that $L(\lambda)$ is an $n \times n$ singular matrix polynomial of degree ℓ and rank r.

Definition 1.7.1 ([42], Def. 2.1)**.** *The* right nullspace*, respectively the* left nullspace*, of* $L(\lambda) \in \mathbb{K}[\lambda]^{n\times n}$ *is the* $\mathbb{K}(\lambda)$*-subspace of* $\mathbb{K}(\lambda)^n$*, respectively of* $\mathbb{K}(\lambda)^{1\times n}$*, defined by*

$$\{x(\lambda) \in \mathbb{K}(\lambda)^n \,;\, L(\lambda)\,x(\lambda) = 0\}, \quad \textit{respectively} \quad \{y(\lambda) \in \mathbb{K}(\lambda)^{1\times n} \,;\, y(\lambda)L(\lambda) = 0\}.$$

It is always possible to construct a basis of the right nullspace, respectively of the left nullspace, of $L(\lambda)$ constituted only of vectors polynomials in the variable λ. Indeed, it suffices to consider an arbitrary basis and to multiply each vector by the least common denominator of its components. Such a basis is called a *right polynomial basis*, respectively *a left polynomial basis*, of $L(\lambda)$ and the sum of the degrees of its elements is called the *order* of the basis. The following definition is a particular case of [42, Def. 2.4].

Definition 1.7.2. *Let* $\mathcal{B}$ *be a right polynomial basis, respectively a left polynomial basis, of* $L(\lambda)$ *and let* δ *be the order of* $\mathcal{B}$*. If* δ *is minimal among the orders of all right polynomial bases, respectively all left polynomial bases, of* $L(\lambda)$ *then* $\mathcal{B}$ *is called a* right minimal basis*, respectively a* left minimal basis*, of* $L(\lambda)$*.*

Example 1.7.1. *Let* $L(\lambda)$ *be the* 4×4 *singular matrix polynomial of degree* 2 *and rank* 2 *given by*

$$L(\lambda) = \begin{pmatrix} \lambda^2 & 1 & \lambda & \lambda^2+\lambda \\ 2 & 0 & 0 & 2 \\ 0 & 1 & \lambda & \lambda \\ \lambda & 1 & \lambda & 2\,\lambda \end{pmatrix}.$$

The two families

$$\mathcal{B}_1 = \left(\begin{pmatrix} 0 \\ \lambda \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ \lambda \\ 0 \\ -1 \end{pmatrix} \right) \quad \textit{and} \quad \mathcal{B}_2 = \left(\begin{pmatrix} 0 \\ \lambda \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ -1 \end{pmatrix} \right)$$

form two right polynomial bases of $L(\lambda)$*. The order of* $\mathcal{B}_1$*, respectively* $\mathcal{B}_2$*, is equal to* 2*, respectively* 1*. Thus* $\mathcal{B}_1$ *is not a right minimal basis of* $L(\lambda)$ *whereas one can show that* $\mathcal{B}_2$ *is a right minimal basis of* $L(\lambda)$ *(see [65, Th. 6.5-10, page 458]).*

The ordered list of the degrees of the elements in any right minimal basis of $L(\lambda)$ does not depend on the choice of the right minimal basis. In other words, if $\mathcal{B}_1 = (x_1(\lambda), \ldots, x_{n-r}(\lambda))$ and $\mathcal{B}_2 = (y_1(\lambda), \ldots, y_{n-r}(\lambda))$ are two right minimal bases of $L(\lambda)$ such that for $1 \leq i \leq j \leq n-r$, one has $\deg(x_i(\lambda)) \leq \deg(x_j(\lambda))$ and $\deg(y_i(\lambda)) \leq \deg(y_j(\lambda))$, then for $i = 1, \ldots, n-r$, $\deg(x_i(\lambda)) = \deg(y_i(\lambda))$ (see [42] and references therein). An analogous result can also be stated for left minimal bases. These degrees are called the *right minimal* (or *Kronecker*) *indices*, respectively the *left minimal* (or *Kronecker*) *indices*, of $L(\lambda)$ and the sum of the right and left minimal indices does not exceed $r\,\ell$ (see for example [104, Corollary 3.1]). Note that the minimal indices of a matrix pencil can be read off from its *Kronecker canonical form* (see [45, Chap. 12] and [65, Chap. 6]). Furthermore, relations between the minimal indices of a matrix polynomial $L(\lambda)$ and those of its linearizations are provided in [42].

Lemma 1.7.1. *Let* $P \in \mathrm{GL}_n(\mathbb{K})$ *be an invertible constant matrix. The right minimal indices of the matrix polynomials* $L(\lambda)$ *and* $L(\lambda)\,P$ *are the same.*

Proof. Consider the endomorphism of the $\mathbb{K}(\lambda)$-vector space $\mathbb{K}(\lambda)^n$ defined by

$$\begin{array}{rccc} P. : & \mathbb{K}(\lambda)^n & \longrightarrow & \mathbb{K}(\lambda)^n \\ & x(\lambda) & \longmapsto & P\,x(\lambda). \end{array}$$

Since P is invertible, this map sends any basis of a $\mathbb{K}(\lambda)$-subspace $\mathcal{V}$ of $\mathbb{K}(\lambda)^n$ to a basis of the $\mathbb{K}(\lambda)$-subspace $P\,\mathcal{V}$ defined by

$$P\,\mathcal{V} = \{P\,x(\lambda)\,;\, x(\lambda) \in \mathcal{V}\}.$$

Moreover, as $P \in \mathrm{GL}_n(\mathbb{K})$, if $x(\lambda) \in \mathbb{K}[\lambda]^n$, then $\deg(x(\lambda)) = \deg(P\,x(\lambda))$. The proof of the lemma follows by taking $\mathcal{V} = \ker(L(\lambda)\,P)$ $(P\,\mathcal{V} = \ker(L(\lambda)))$. □

The computation of a minimal basis of a singular matrix polynomial $L(\lambda)$ can be reduced to the computation of a minimal approximant basis of $L(\lambda)$ viewed as a formal power series matrix.

Minimal approximant bases (see [62, 103]). Let $F(\lambda)$ be an $m \times n$ matrix whose entries are formal power series over $\mathbb{K}$ and let $\sigma \in \mathbb{N}$. A vector polynomial $v(\lambda) \in \mathbb{K}[\lambda]^n$ satisfying

$$F(\lambda)\,v(\lambda) = 0 \qquad \mod \lambda^\sigma \tag{1.14}$$

is said *to have order* (F, σ). The set $\mathcal{M}_{F,\sigma}$ of all vector polynomials of order (F, σ) is a free $\mathbb{K}[\lambda]$-module of rank n (see [96]).

Definition 1.7.3 ([62]). *A basis* $(v_1(\lambda), \ldots, v_n(\lambda))$ *of* $\mathcal{M}_{F,\sigma}$ *with ordered degrees* $\deg(v_1) \leq \cdots \leq \deg(v_n)$ *is called a* minimal approximant basis for $F(\lambda)$ of order σ *if any other basis of* $\mathcal{M}_{F,\sigma}$ *with degrees* $d_1 \leq \cdots \leq d_n$ *satisfies* $d_i \geq \deg(v_i)$ *for* $i = 1, \ldots, n$.

Given a singular matrix polynomial $L(\lambda)$, it is easy to see that the right nullspace of $L(\lambda)$ is contained in the module $\mathcal{M}_{L,\sigma}$ for any positive integer σ. However, in order to extract a minimal basis of $L(\lambda)$ from a minimal approximant basis for $L(\lambda)$ of order σ, we have to choose an appropriate value for σ. Keeping in mind that the degrees of the vectors in a right minimal basis of a matrix polynomial of rank r and degree ℓ do not exceed $r\,\ell$, we deduce from [62, Th. 2.4] that:

Theorem 1.7.1. *Let* $L(\lambda) \in \mathbb{K}[\lambda]^{n \times n}$ *be a singular matrix polynomial of rank* r *and degree* ℓ. *Let* $\mathcal{B}$ *be a minimal approximant basis for* $L(\lambda)$ *of order* $\sigma = (r+1)\,\ell + 1$. *Then, the* $(n - r)$ *elements of* $\mathcal{B}$ *of minimal degrees form a right minimal basis of* $L(\lambda)$.

The complexity of computing a minimal approximant basis of order σ for an $n \times n$ matrix of formal power series is $O^\sim(n^\omega\,\sigma)$ operations in $\mathbb{K}$ (see [62, 103]). From Theorem 1.7.1, we obtain:

Lemma 1.7.2. *The cost of computing a minimal basis of an* $n \times n$ *singular matrix polynomial of degree* ℓ *is at most* $O^\sim\left(n^{\omega+1}\,\ell\right)$ *operations in* $\mathbb{K}$.

Note that for our implementation in Maple, we use the command *MinimalBasis* of the MatrixPolynomialAlgebra package.

Chapter 2

Symbolic Methods for Computing Regular Solutions of Higher-Order Linear Differential Systems of the First Kind

Contents

2.1 Introduction

We consider systems of linear differential equations of arbitrary order $\ell \geq 1$ with meromorphic coefficients and treat the problem of computing their formal solutions at a singular point x_0 which can be supposed, without any loss of generality, located at the origin. Since we are interested in the local analysis, we can hence assume that the coefficients of the systems considered are formal series.

In the sequel, for ease of presentation, we will use the *Euler derivation* $\vartheta = x\frac{d}{dx}$ instead of the standard derivation $\frac{d}{dx}$. These derivations are related by

$$\forall\, i \geq 1, \quad x^i \left(\frac{d}{dx}\right)^i = \vartheta\,(\vartheta - 1)\cdots(\vartheta - i + 1)\,.$$

For a subfield $\mathbb{K}$ of $\mathbb{C}$, we denote by $\mathbb{K}[[x]]$ the ring of formal power series over $\mathbb{K}$ in the variable x and by $\mathbb{K}((x))$ its field of fractions. One has $\mathbb{K}((x)) = \mathbb{K}[[x]][x^{-1}]$.

We consider a system of n linear differential equations of order $\ell \geq 1$ of the form

$$\mathcal{L}(x,\vartheta)(y(x)) = A_\ell(x)\,\vartheta^\ell(y(x)) + \cdots + A_1(x)\,\vartheta(y(x)) + A_0(x)\,y(x) = 0, \tag{2.1}$$

where for $i = 0,\ldots,\ell$, $A_i(x)$ is an $n \times n$ matrix with entries in $\mathbb{K}[[x]]$ and $y(x) = (y_1(x),\ldots,y_n(x))^T$ is an unknown n-dimensional vector. In this chapter, we assume that $A_\ell(0)$ is an invertible matrix which is equivalent to assuming that the *leading coefficient matrix* $A_\ell(x)$ is invertible over $\mathbb{K}[[x]]$.

The classical approach to deal with such systems consists in converting them into a first-order system of size $n\,\ell$

$$\vartheta(Y(x)) = \mathcal{C}(x)\,Y(x), \tag{2.2}$$

where

$$Y(x) = \begin{pmatrix} y(x) \\ \vartheta(y(x)) \\ \vdots \\ \vartheta^{\ell-2}(y(x)) \\ \vartheta^{\ell-1}(y(x)) \end{pmatrix} \quad \text{and} \quad \mathcal{C}(x) = \begin{pmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ \widetilde{A}_0(x) & \widetilde{A}_1(x) & \widetilde{A}_2(x) & \ldots & \widetilde{A}_{\ell-1}(x) \end{pmatrix} \tag{2.3}$$

is the block companion matrix with $\widetilde{A}_i(x) = -A_\ell^{-1}(x)\,A_i(x) \in \mathbb{K}[[x]]^{n\times n}$ for $i = 0,\ldots,\ell-1$. Since $\mathcal{C}(0) \neq 0$, the first-order system (2.2) is of the first kind (see [8, Chap. 2], [37, Chap. 4] or [101, Chap. 2]) and hence we will say that System (2.1) is of the *first kind* as well. It is well-known (see for instance [8, Chap. 2], [37, Chap. 4] or [101, Chap. 2]) that a fundamental system of solutions of System (2.2) is provided by the columns of a matrix of the form

$$W(x) = \Phi(x)\,x^{\mathcal{J}},$$

where $\Phi(x) \in \overline{\mathbb{K}}[[x]]^{n\,\ell\times n\,\ell}$ and $\mathcal{J} = \operatorname{diag}(J_1,\ldots,J_r) \in \overline{\mathbb{K}}^{n\,\ell\times n\,\ell}$ with J_i a Jordan block of the form

$$J_i = \begin{pmatrix} \lambda_i & 1 & & (0) \\ & \ddots & \ddots & \\ & & & 1 \\ & (0) & & \lambda_i \end{pmatrix} \qquad (\lambda_i \in \overline{\mathbb{K}}).$$

Therefore, a fundamental system of solutions of System (2.1) is given by the columns of the matrix

$$U(x) = \mathcal{P}\,W(x) = \mathcal{P}\,\Phi(x)\,x^{\mathcal{J}} = Q(x)\,x^{\mathcal{J}} = Q(x)\operatorname{diag}\left(x^{J_1},\ldots,x^{J_r}\right),$$

where $\mathcal{P} = \begin{pmatrix} I_n & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{K}^{n\times n\,\ell}$, $Q(x) = \mathcal{P}\,\Phi(x) \in \overline{\mathbb{K}}[[x]]^{n\times n\,\ell}$ and

$$x^{J_i} = \begin{pmatrix} x^{\lambda_i} & x^{\lambda_i}\log(x) & \cdots & x^{\lambda_i}\frac{\log^{k_i-1}(x)}{(k_i-1)!} \\ & \ddots & \ddots & \vdots \\ & (0) & \ddots & x^{\lambda_i}\log(x) \\ & & & x^{\lambda_i} \end{pmatrix}$$

with k_i being the size of the Jordan block J_i. We thus get k_i linearly independent solutions of System (2.1) which can be written of the form

$$\begin{cases} y_{i,1}(x) = x^{\lambda_i}\, Q(.,N_i+1), \\ y_{i,2}(x) = x^{\lambda_i}\, Q(.,N_i+2) + x^{\lambda_i}\, \log(x)\, Q(.,N_i+1), \\ \qquad\qquad \vdots \\ y_{i,k_i}(x) = x^{\lambda_i}\, Q(.,N_i+k_i) + x^{\lambda_i}\, \log(x)\, Q(.,N_i+k_i-1) + \cdots + x^{\lambda_i}\, \frac{\log^{k_i-1}(x)}{(k_i-1)!}\, Q(.,N_i+1), \end{cases}$$

where $Q(.,j)$ denote the jth column of the matrix Q and $N_i = \sum_{j=1}^{i-1} k_j$. Consequently, the formal solution space of System (2.1) is spanned by $n\,\ell$ linearly independent formal solutions of the form

$$y(x) = x^{\lambda_0}\, z(x), \tag{2.4}$$

where $\lambda_0 \in \overline{\mathbb{K}}$ and $z(x) \in \overline{\mathbb{K}}[[x]][\log(x)]^n$ with degree in $\log(x)$ less than $n\,\ell$. These solutions are known as *regular formal solutions*. More generally, a linear combination of this type of solutions is called a *regular formal solution* and the vector space generated by the regular formal solutions is known as the *regular formal solution space*. In this chapter, we address the problem of constructing directly a basis of the regular formal solution space of System (2.1) by avoiding the transformation into first-order systems. To achieve this, we propose two methods:

- The first one is a generalization of the approach developed by Poole in [86, Chap. 5, § 16] for the scalar case ($n = 1$). It consists in arranging a regular solution $y(x)$ of the form (2.4) as a series in x whose coefficients are polynomials in $t = \log(x)$ ($\vartheta = \frac{d}{dt}$) of degree less than $n\,\ell$, that is,

$$y(x) = x^{\lambda_0} \left(U_0(t) + U_1(t)\, x + \cdots + U_i(t)\, x^i + \cdots\right), \tag{2.5}$$

  with $\lambda_0 \in \overline{\mathbb{K}}$, $\forall\ m \geq 0$, $U_m(t) \in \overline{\mathbb{K}}[t]^n$ and $U_0 \neq 0$. In the sequel, we will refer to $y(x)$ given by (2.5) as a *regular solution of exponent* λ_0. Plugging $y(x)$ into System (2.1), one finds that λ_0 and $U_0(t)$ must satisfy

$$L_0(\vartheta)\left(x^{\lambda_0}\, U_0\right) = 0,$$

  where $L_0(\lambda)$ designates the matrix polynomial

$$L_0(\lambda) = A_\ell(0)\, \lambda^\ell + \cdots + A_1(0)\, \lambda + A_0(0)$$

  whose determinant will play the same role as the *indicial polynomial* in the scalar case. It follows that λ_0 must be chosen as an eigenvalue of the matrix polynomial $L_0(\lambda)$ and the coefficients of the polynomial U_0 form a Jordan chain for $L_0(\lambda)$ associated with λ_0 (see Lemma 2.4.1 on page 42). For $m \geq 1$, we find

$$L_0(\vartheta + \lambda_0 + m)\,(U_m) = P_m(t),$$

  where $P_m(t)$ depends on $U_0(t), \ldots, U_{m-1}(t)$. The problem is thus reduced to computing polynomial solutions in t of non-homogeneous linear differential systems with constant coefficients.

- The second one is a generalization of Frobenius' method, a well-known approach for computing regular solutions of scalar linear differential equations (see [10, Chap. 1], [37, Chap. 4, Section 8], [44] or [59, Chap. 16]). This approach consists in computing the regular solutions of System (2.1) from the (logarithm-free) formal power series solutions of non-homogeneous linear differential systems well-defined.

The chapter is organized as follows. In Section 2.2, we consider linear differential systems of first-order. We recall the classification of singularities and the classical algorithm for computing a fundamental solution matrix of first-order linear differential systems of the first-kind. In Section 2.3, we review the existing methods for handling higher-order systems of the form (2.1). In Section 2.4, we examine systems of the form (2.1) with constant coefficient matrices A_i. We show how the regular solutions of such systems are strongly connected to the notion of Jordan chains of matrix polynomials. In Section 2.5, we show that non-homogeneous differential systems with constant coefficients and polynomial right-hand side in $t = \log(x)$ always admit a polynomial solution in t and we give a bound on its degree and an algorithm for its computation. In Section 2.6, we propose a direct method inspired by Poole's approach [86] for handling systems of the form (2.1). In Section 2.7, we generalize Frobenius' method to compute regular solutions of systems of the form (2.1). Finally, in Section 2.8, we present some tables of timings.

The content of this chapter is published in [16] and in a part of [17, 18].

2.2 Local analysis of first-order linear differential systems

Consider a system of ordinary differential equations of first-order of the form

$$y'(x) = A(x)\, y(x) \tag{2.6}$$

where $A(x)$ is an $n \times n$ complex-valued matrix, analytic and single-valued in a neighborhood of the point $x = 0$, say $0 < |x| < a$ (a being a positive real number), having at most an isolated singularity at $x = 0$, $y(x)$ is an unknown n-dimensional vector and $y'(x) = \frac{dy}{dx}(x)$. The point $x = 0$ is called a *singular point* or a *singularity* for System (2.6) if $A(x)$ has a singularity at $x = 0$. Otherwise, it is called an *ordinary point*.

2.2.1 Classification of singularities

A *fundamental solution matrix* of System (2.6) is defined as an $n \times n$ matrix whose columns form n linearly independent solutions of (2.6). It has been shown in [37, Chap. 4] that any fundamental solution matrix of (2.6) can be written in the form

$$W(x) = \Phi(x)\, x^D,$$

where $\Phi(x)$ is single-valued, analytic in $0 < |x| < a$ and D is a constant matrix. If the point $x = 0$ is an ordinary point for (2.6), then $\Phi(x)$ is also analytic at $x = 0$. If $x = 0$ is a singularity for System (2.6), then

Definition 2.2.1. *The point $x = 0$ is called a* regular singularity *for System (2.6) if $\Phi(x)$ has at most a pole at $x = 0$. Otherwise, it is called an* irregular singularity.

This classification of the singularity building on the knowledge of a fundamental solution matrix is not immediately checkable for a given system of the form (2.6).

Suppose now that $A(x)$ has at most a pole at $x = 0$. A second classification of the singularity based on the nature of the pole $x = 0$ of $A(x)$ is provided. Indeed, write matrix $A(x)$ of System (2.6) as

$$A(x) = \frac{1}{x^{p+1}} \sum_{i=0}^{\infty} A_i\, x^i,$$

where $A_i \in \mathbb{C}^{n\times n}$, $A_0 \neq 0$ and $p \in \mathbb{N}$. The integer p is called the *Poincaré-rank* of System (2.6).

Definition 2.2.2. *The point $x = 0$ is called a* singularity of the first kind *for System (2.6) if $p = 0$, or in other terms, $x = 0$ is a simple pole for $A(x)$. Otherwise, it is called a* singularity of the second kind.

These two classifications of the singularity $x = 0$ are not directly connected. However, it has been shown in [37, Chap. 4, Th. 2.1] that if the point $x = 0$ is a singularity of the first kind for (2.6), then it is a regular singularity for (2.6). The converse is not always true and a singularity of the second kind for (2.6) can be a regular singularity.

A transformation of the form

$$y(x) = T(x)\, z(x)$$

with $T(x)$ an invertible matrix with meromorphic entries is known as *gauge transformation* and transforms System (2.6) into a new system $z'(x) = B(x)\, z(x)$, where

$$B(x) = T^{-1}(x)\left(A(x)\, T(x) - T'(x)\right)$$

is often denoted by $T[A]$. The matrices A and B or the corresponding systems are said to be *gauge-equivalent*. It has been proven by Moser in [77] that, in the case where the point $x = 0$ is a regular singularity, there exists a polynomial gauge transformation $T(x)$ such that $T[A]$ has a simple pole at $x = 0$ (*i.e.*, $x = 0$ is a singularity of the first kind for the system defined by $T[A]$). More precisely, Moser developed an algorithm (see also [12, 54, 56]) which returns a system gauge-equivalent to (2.6) having minimal Poincaré-rank among all the gauge-equivalent systems . Using this algorithm, one can then determine the nature of the singularity (regular or irregular) for a given system.

Recall that by means of the so-called "cyclic vector" method (see [39, 11, 35]), any system of the form (2.6) can be reduced to an equivalent scalar linear differential equation of order n. In other words, one can always construct a gauge transformation $T(x) \in \mathrm{GL}_n(\mathbb{C}((x)))$ such that

$$T[A] = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0(x) & -a_1(x) & -a_2(x) & \dots & -a_{n-1}(x) \end{pmatrix}$$

with $a_i(x) \in \mathbb{C}((x))$. Hence, System (2.6) is equivalent to the scalar linear differential equation

$$z^{(n)}(x) + a_{n-1}(x)\, z^{(n-1)}(x) + \cdots + a_0(x)\, z(x) = 0, \tag{2.7}$$

where $y(x)$ and $z(x)$ are related by $y(x) = T(x)\,(z(x), z'(x), \dots, z^{(n-1)}(x))^T$. While it is not always obvious whether a differential system has a regular or irregular singularity at $x = 0$, it is much easier to check it for a scalar linear differential equation. This is provided by the Fuchs criterion which establishes a link between the nature of the singularity for (2.7) and the valuation at $x = 0$ of its coefficients $a_0(x), \dots, a_{n-1}(x)$; indeed, $x = 0$ is a regular singularity for (2.7) if and only if for $i = 0, \dots, n-1$, $a_i(x)$ has a pole at $x = 0$ of order less than or equal to $n - i$. The reduction to scalar linear differential equations constitutes then another approach for the classification of the singularities of systems of the form (2.6). Nevertheless, it is important to mention that computing an equivalent scalar equation from a given system can be very costly specially when the system has large size n, and the coefficients of the obtained equation can be very huge (see [36]).

2.2.2 Computation of a fundamental solution matrix for systems of the first kind

Consider a first-order linear differential system of the first kind of the form

$$y'(x) = A(x)\,y(x) \quad \text{with } A(x) = \frac{1}{x}\sum_{i=0}^{\infty} A_i\,x^i \in x^{-1}\mathbb{K}[[x]]^{n\times n} \quad \text{and } A_0 \neq 0. \tag{2.8}$$

The aim of this subsection is to recall the classical method for computing a formal fundamental solution matrix for systems of the first kind (2.8).

Suppose first that the matrix $A(x)$ has *good spectrum* (see [8, page 18]), *i.e.*, the distinct eigenvalues of A_0 do not differ by integers. This implies that the matrices $A_0 + i\,I$ and A_0 have disjoint spectrums for every nonzero integer i. It follows the existence of a gauge transformation $P(x) = \sum_{i=0}^{\infty} P_i\,x^i \in \mathrm{GL}_n(\mathbb{K}[[x]])$ with $P_0 = I$ that transforms System (2.8) into the system

$$z'(x) = \frac{A_0}{x}\,z(x)$$

whose general solution is given by $C\,x^{A_0}$, where $C \in \mathrm{GL}_n(\mathbb{K})$. To compute the coefficients P_i of $P(x)$ for $i \geq 1$, we have

$$\begin{aligned} P[A] = \frac{A_0}{x} &\iff A(x)\,P(x) - P'(x) = P(x)\frac{A_0}{x} \\ &\iff P_i\,(A_0 + i\,I) - A_0\,P_i = \sum_{j=0}^{i-1} A_{i-j}\,P_j, \quad \forall\, i \geq 1. \end{aligned}$$

Since $A_0 + i\,I$ and A_0 have disjoint spectrums, the above system has a unique solution (see for example [60]). Thus, the coefficients P_i for $i \geq 1$ are uniquely determined from the coefficients of $A(x)$. Consequently, a formal fundamental solution matrix of System (2.8) with good spectrum is given by $P(x)\,x^{A_0}$ (see [37, Chap. 4, Th. 4.1] and [101, Chap. 2, Th. 5.5]).

Suppose now that $A(x)$ does not have a good spectrum. The idea is to seek a gauge transformation $T(x)$ such that $T[A]$ has a good spectrum. For achieving this, one can proceed as follows. Suppose that A_0 has exactly r distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_r$ such that $\lambda_i - \lambda_1 = n_i \in \mathbb{N}^*$ for $i = 2, \ldots, s$ $(2 \leq s \leq r)$ and $\lambda_i - \lambda_1 \notin \mathbb{Z}$ for $i = s+1, \ldots, r$. Let m_i denote the (algebraic) multiplicity of eigenvalue λ_i for $i = 2, \ldots, s$. Let P be an invertible matrix such that $P^{-1}A_0\,P$ is of the form $\mathrm{diag}(M, N)$ where M is of size $m_1 \times m_1$ having λ_1 as the unique eigenvalue. Let $S(x) = \mathrm{diag}(I_{m_1}, x\,I_{n-m_1})$. The gauge transformation $T_1(x) = PS(x)$ transforms System (2.6) into another one $z'(x) = \frac{A^{[1]}(x)}{x}\,z(x)$ with $A^{[1]}(x) \in \mathbb{K}[[x]]^{n\times n}$ such that the eigenvalues of $A^{[1]}(0)$ are $\lambda_1, \lambda_2 - 1, \ldots, \lambda_r - 1$ (see [37, Chap. 4, Sec. 4] or [54, Chap. 4]). Hence, repeating this process n_2 times, we find that the eigenvalues of the matrix $A^{[n_2]}(0)$ are $\lambda_1, \lambda_2 - n_2 = \lambda_1, \ldots, \lambda_r - n_2$ and λ_1 is now of multiplicity equal to $m_1 + m_2$. Hence, after a finite number of iterations, we obtain a gauge transformation $T(x) \in \mathbb{K}[x]^{n\times n}$ invertible for $x \neq 0$ such that $T[A]$ has a good spectrum.

Theorem 2.2.1 ([37], Chap. 4, Th. 4.2)**.** *Any system of the first kind of the form (2.8) has a formal fundamental solution matrix of the form $\Phi(x)\,x^{\Lambda}$ where $\Phi(x) \in \mathbb{K}[[x]]^{n\times n}$ is invertible over $\mathbb{K}((x))$ and $\Lambda \in \mathbb{K}^{n\times n}$ whose eigenvalues do not differ by integers.*

2.3 Existing methods for computing regular solutions

As we have already mentioned, the classical method for solving higher-order linear differential systems consists in transforming them into first-order systems of the form (2.2). We have exposed

in Subsection 2.2.2 a method for computing a formal fundamental solution matrix for first-order systems of the first-kind. There exists another algorithm due to Barkatou and Pflügel [21][1] which computes a basis of the regular formal solution space of first-order systems even in the case of an irregular singularity. An implementation of this algorithm can be found in the ISOLDE package of MAPLE [23]. However, the conversion into first-order systems has the computational drawback of increasing the size of the system. Furthermore, the algorithms presented for first-order systems do not take advantage of the structure of the block companion matrix $\mathcal{C}(x)$ given by (2.3). For those reasons, the purpose of this chapter is to look for methods that handle directly Systems of the form (2.1).

To our knowledge, there exist only few works on that direction. In [5], Abramov et al. prove that Heffter's method [53, Chap. 8], known for solving scalar linear differential equations, can be generalized for higher-order systems with polynomial coefficients not necessarily of the first kind; the problem of finding regular solutions is reduced to computing Laurent series solutions of auxiliary non-homogeneous higher-order linear differential systems with polynomial coefficients. To compute these Laurent series solutions, the authors transform first the non-homogeneous differential systems into matrix recurrence equations (see [2]) using the dictionary: $x \longmapsto E^{-1}$ and $\frac{d}{dx} \longmapsto (n+1)\,E$ where E is the shift operator $E : n \longmapsto n+1$. Then, they solve these matrix recurrence equations using the EG-elimination method (see [1, 4]). The EG-elimination method transforms a matrix recurrence equation into another one whose leading coefficient matrix (or trailing coefficient matrix) in nonsingular. In the case of systems of the first-kind, there is no need for the EG-elimination since the leading coefficient matrix is already nonsingular. The LINEARFUNCTIONALSYSTEMS package of MAPLE contains an implementation of this method.

In [64], Jódar and Legua consider systems of the form (2.1) with $\ell = 2$ and $A_2(x) = I_n$ and generalize the Frobenius method to such systems under some restrictions, such as supposing that the matrix $\mathcal{C}(0)$, where $\mathcal{C}(x)$ is given by (2.3), is diagonalizable. Therefore, the regular formal solution space of such systems is only composed of logarithm-free regular solutions, *i.e.*, solutions of the form $y(x) = x^{\lambda_0}\, z(x)$ with $\lambda_0 \in \overline{\mathbb{K}}$ and $z(x) \in \overline{\mathbb{K}}[[x]]^n$.

In [79], the authors consider particular systems called *Bessel matrix differential equations*. These systems are of the form (2.1) with $\ell = 2$, $A_2(x) = I_n$, $A_1(x) = 0$ and $A_0(x) = I_n\, x^2 - A^2$ (A being a constant matrix). Again, to compute the regular solutions of such systems, they impose some conditions on the matrix A. Later in [63], they consider another type of systems of second-order and introduce the class of Laguerre matrix polynomials which appears as finite series solutions of such systems. Finally, in [80], the authors develop a Frobenius matrix method for solving systems of the form (2.1). They look for solutions of the form $\left(\sum_{i=0}^{\infty} D_i\, x^i\right) x^Z$ where $D_i \in \overline{\mathbb{K}}^{n\times m}$ and $Z \in \overline{\mathbb{K}}^{m\times m}$. For computing such solutions, the authors assume that the eigenvalues of the matrix $\mathcal{C}(0)$, where $\mathcal{C}(x)$ is given by (2.3), do not differ by integers.

2.4 Euler's matrix differential equations

In this section, we investigate the case where the coefficient matrices of System (2.1) are constant, *i.e.*, we consider systems of the form

$$A_\ell\, \vartheta^\ell(y(x)) + \cdots + A_1\, \vartheta(y(x)) + A_0\, y(x) = 0, \tag{2.9}$$

where for $i = 0, \ldots, \ell$, $A_i \in \mathbb{K}^{n\times n}$ and $A_\ell \in \mathrm{GL}_n(\mathbb{K})$. In the scalar case, that is, when we have a differential equation of the form

$$a_\ell\, \vartheta^\ell(y(x)) + \cdots + a_1\, \vartheta(y(x)) + a_0\, y(x) = 0,$$

[1] We will review this algorithm in more details in the next chapter.

where for $i = 0, \ldots, \ell$, $a_i \in \mathbb{K}$ and $a_\ell \neq 0$, such an equation is called *Euler's equation* (see [37, Chap. 4, Sec. 5]). A basis of the regular formal solution space of an Euler equation is composed of ℓ regular solutions $y(x) = x^{\lambda_0}\, z(x)$, where $\lambda_0 \in \overline{\mathbb{K}}$ is a root of the so-called *indicial polynomial* defined by $f(\lambda) = a_\ell\, \lambda + \cdots + a_1\, \lambda + a_0$ and $z(x) \in \overline{\mathbb{K}}[\log(x)]$ is of degree in $\log(x)$ less than the multiplicity of λ_0 as a root of $f(\lambda)$ (see [37, Chap. 4]). In the sequel, we will refer to a system of the form (2.9) as *Euler's matrix differential equation* and we will show that the result obtained in the scalar case can be extended to the matrix case. For this, define the matrix polynomial $L(\lambda)$ associated with System (2.9) by

$$L(\lambda) = A_\ell\, \lambda^\ell + \cdots + A_1\, \lambda + A_0. \tag{2.10}$$

The following lemma shows that the determinant of $L(\lambda)$ plays the same role as the indicial polynomial in the scalar case.

Lemma 2.4.1. *Consider an Euler matrix differential equation of the form (2.9). The vector-valued function*

$$y(x) = x^{\lambda_0} \left(v_{k-1} + v_{k-2} \frac{\log(x)}{1!} + \cdots + v_0 \frac{\log^{k-1}(x)}{(k-1)!} \right),$$

where $\lambda_0 \in \overline{\mathbb{K}}$, $k \in \mathbb{N}^*$ and $v_0, \ldots, v_{k-1} \in \overline{\mathbb{K}}^n$, is a solution of (2.9) if and only if λ_0 is an eigenvalue of $L(\lambda)$ defined in (2.10) and $v_0, \ldots, v_{k-1}$ form a Jordan chain for $L(\lambda)$ associated with λ_0.

Proof. Write the Taylor series expansion for the matrix polynomial $L(\lambda)$ at $\lambda = \lambda_0$:

$$L(\lambda) = L(\lambda_0) + \frac{1}{1!} L'(\lambda_0)(\lambda - \lambda_0) + \cdots + \frac{1}{j!} L^{(j)}(\lambda_0)(\lambda - \lambda_0)^j + \cdots + \frac{1}{\ell!} L^{(\ell)}(\lambda_0)(\lambda - \lambda_0)^\ell.$$

We thus have

$$\begin{aligned} L(\vartheta)(y(x)) = L(\lambda_0)\, y(x) + \frac{1}{1!} L'(\lambda_0)(\vartheta - \lambda_0)(y(x)) + \cdots \\ + \frac{1}{j!} L^{(j)}(\lambda_0)(\vartheta - \lambda_0)^j(y(x)) + \cdots + \frac{1}{\ell!} L^{(\ell)}(\lambda_0)(\vartheta - \lambda_0)^\ell(y(x)). \end{aligned} \tag{2.11}$$

It is easy now to check that $(\vartheta - \lambda_0)^j(y(x))$ with $y(x) = x^{\lambda_0} \sum_{i=0}^{k-1} v_{k-i-1} \frac{\log^i(x)}{i!}$ is equal to

$$(\vartheta - \lambda_0)^j(y(x)) = \begin{cases} x^{\lambda_0} \left(v_{k-1-j} + v_{k-2-j} \frac{\log(x)}{1!} + \cdots + v_0 \frac{\log^{k-1-j}(x)}{(k-1-j)!} \right) & \text{if} \quad j \leq k-1, \\ 0 & \text{if} \quad j \geq k. \end{cases}$$

Hence, substituting $y(x) = x^{\lambda_0} \sum_{i=0}^{k-1} v_{k-i-1} \frac{\log^i(x)}{i!}$ in (2.11) and identifying the coefficients of $\frac{\log^i(x)}{i!}$, for $i = 0, \ldots, k-1$, to zero yield

$$\sum_{j=0}^{p} \frac{1}{j!} L^{(j)}(\lambda_0)\, v_{p-j} = 0 \quad \text{for } p = 0, \ldots, k-1,$$

which ends the proof. □

Proposition 2.4.1. *Consider the Euler matrix differential equation given by (2.9). For each eigenvalue λ_0 of $L(\lambda)$ defined in (2.10), let $v_{\lambda_0,i,0}, \ldots, v_{\lambda_0,i,\kappa_i(\lambda_0)-1}$ for $i = 1, \ldots, m_g(\lambda_0)$ form a canonical set of Jordan chains for $L(\lambda)$ corresponding to λ_0 (the $\kappa_i(\lambda_0)$'s being the partial multiplicities at λ_0). For $i = 1, \ldots, m_g(\lambda_0)$ and $j = 0, \ldots, \kappa_i(\lambda_0) - 1$, define*

$$y_{\lambda_0,i,j}(x) = x^{\lambda_0} \sum_{k=0}^{j} v_{\lambda_0,i,j-k} \frac{\log^k(x)}{k!}.$$

The vector-valued functions $y_{\lambda_0,i,j}(x)$ *with* $\lambda_0 \in \sigma(L)$, $i = 1, \ldots, m_g(\lambda_0)$ *and* $j = 0, \ldots, \kappa_i(\lambda_0)-1$ *form a basis of the regular formal solution space of System (2.9).*

Proof. According to Lemma 2.4.1, each vector-valued function $y_{\lambda_0,i,j}$ with $\lambda_0 \in \sigma(L)$, $i \in \{1, \ldots, m_g(\lambda_0)\}$ and $j \in \{0, \ldots, \kappa_i(\lambda_0) - 1\}$ is a solution of System (2.9). Let us show now the linearly independency of these vectors. For $i \in \{1, \ldots, m_g(\lambda_0)\}$, the vectors y_{λ_0,i,j_1} and y_{λ_0,i,j_2} with $j_1 \neq j_2$ are linearly independent over $\overline{\mathbb{K}}$ since they are of different degrees in $\log(x)$. The vectors y_{λ_0,i_1,j_1} and y_{λ_0,i_2,j_2} with $i_1 \neq i_2$ are linearly independent over $\overline{\mathbb{K}}$ since the eigenvectors $v_{\lambda_0,i_1,0}$ and $v_{\lambda_0,i_2,0}$ are so (see Section 1.5 of Chapter 1). Thus, the $y_{\lambda_0,i,j}(x)$ with $\lambda_0 \in \sigma(L)$, $i = 1, \ldots, m_g(\lambda_0)$ and $j = 0, \ldots, \kappa_i(\lambda_0) - 1$ constitute $\sum_{\lambda_0 \in \sigma(L)} m_a(\lambda_0) = \deg(\det(L(\lambda))) = n\,\ell$ (since $A_\ell \in \mathrm{GL}_n(\mathbb{K})$) linearly independent regular solutions of (2.9). Consequently, they form a basis of the regular formal solution space of (2.9) since its dimension is equal to $n\,\ell$. □

We sketch the above proposition in the following algorithm.

ALGORITHM **Euler_MDE**

INPUT: The matrices A_i defining System (2.9).
OUTPUT: A basis of the regular formal solution space of System (2.9).

INITIALIZATION: Let $L(\lambda) = \sum_{i=0}^{\ell} A_i\,\lambda^i$ and Sol = { };

1. Compute $\sigma(L)$; //See Remark 1.1.1 of Chapter 1
2. **For** each element λ_0 of $\sigma(L)$ **do**
   2.1. Compute a canonical set of Jordan chains $v_{i,0}, \ldots, v_{i,\kappa_i(\lambda_0)-1}$, $i = 1, \ldots, m_g(\lambda_0)$, for $L(\lambda)$ associated with λ_0;
   2.2. **For** i from 1 to $m_g(\lambda_0)$ **do**
      For j from 0 to $\kappa_i(\lambda_0) - 1$ **do**
         Let Sol = Sol $\cup \left\{ x^{\lambda_0} \sum_{k=0}^{j} v_{i,j-k} \frac{\log^k(x)}{k!} \right\}$;
      end do;
   end do;
end do;
3. **Return** Sol;

Proposition 2.4.2. *Algorithm* **Euler_MDE** *returns a basis of the regular formal solution space of System (2.9) using at most* $O\big(n^5\,\ell^4\big)$ *arithmetic operations in* $\mathbb{K}$.

Proof. Using Algorithm **CanonicalSet_JC** developed in Chapter 1, Step 2.1 can be performed using at most $O\big(n^4\,\ell^3\,m_a(\lambda_0)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$ (see Remark 1.6.2 of Chapter 1). Since $\sum_{\lambda_0 \in \sigma(L)} m_a(\lambda_0)\,d_{\lambda_0} = \deg(\det(L(\lambda))) = n\,\ell$ (see Remark 1.1.1 of Chapter 1), the above algorithm returns a basis of the regular formal solution space of System (2.9) using at most $O\big(n^5\,\ell^4\big)$ arithmetic operations in $\mathbb{K}$. □

Example 2.4.1. *We consider the linear differential system of order* 3

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \vartheta^3(y(x)) + \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \vartheta^2(y(x)) + \begin{pmatrix} -2 & -1 \\ -2 & -1 \end{pmatrix} \vartheta(y(x)) + \begin{pmatrix} 0 & -1 \\ 0 & -1 \end{pmatrix} y(x) = 0. \qquad (2.12)$$

The matrix polynomial $L(\lambda)$ *associated with this system is given by*

$$L(\lambda) = \begin{pmatrix} \lambda^3 + \lambda^2 - 2\lambda & \lambda^3 + \lambda^2 - \lambda - 1 \\ \lambda^3 + \lambda^2 - 2\lambda & 2\lambda^3 - \lambda - 1 \end{pmatrix}$$

and has three eigenvalues -2, 0 *and* 1 *of algebraic multiplicity* $m_a(-2) = 1$, $m_a(0) = 3$ *and* $m_a(1) = 2$. *Using Algorithm* **CanonicalSet_JC** *developed in Chapter 1, we compute a canonical set of Jordan chains associated respectively with the eigenvalues* -2, 0 *and* 1. *We find respectively*

$$\begin{pmatrix}1\\0\end{pmatrix} \quad (\textit{one Jordan chain of length } 1),$$

$$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}, \begin{pmatrix}0\\-1\end{pmatrix} \quad (\textit{one Jordan chain of length } 3),$$

and

$$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix} \quad (\textit{two Jordan chains of length } 1).$$

Thus, a basis of the regular formal solution space of System (2.12) is given by

$$\begin{pmatrix}x^{-2}\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}\log(x)\\-2\end{pmatrix}, \begin{pmatrix}\frac{\log^2(x)}{2}\\-2\log(x)-1\end{pmatrix}, \begin{pmatrix}x\\0\end{pmatrix}, \begin{pmatrix}0\\x\end{pmatrix}.$$

■

2.5 Non-homogeneous linear differential systems with constant coefficients

As mentioned in the introduction of this chapter, Poole's method reduces the problem of computing regular formal solutions of System (2.1) to the one of computing polynomial solutions in $t = \log(x)$ of non-homogeneous linear differential systems with constant coefficients and a right-hand side polynomial in t. In this section, we show that such a system is always consistent and we give a method for computing its general polynomial solution in t.

2.5.1 Existence of polynomial solutions in $t = \log(x)$

In the following proposition, we prove that non-homogeneous linear differential systems with constant coefficients and a right-hand side polynomial in $t = \log(x)$ admit at least one polynomial solution in t whose degree can be bounded explicitly.

Proposition 2.5.1. *Let* $t = \log(x)$ *and consider the non-homogeneous linear differential system*

$$A_\ell\, \vartheta^\ell(y(x)) + \cdots + A_1\, \vartheta(y(x)) + A_0\, y(x) = \phi(t), \tag{2.13}$$

where for $i = 0, \ldots, \ell$, $A_i \in \mathbb{F}^{n\times n}$ *(*$\mathbb{F}$ *being a subfield of* $\mathbb{C}$*),* $A_\ell \in \mathrm{GL}_n(\mathbb{F})$ *and* $\phi(t)$ *is an* n*-dimensional vector polynomial in* t *of degree* d *with coefficients in* $\mathbb{F}$. *Let* $L(\lambda)$ *be the matrix polynomial defined by*

$$L(\lambda) = A_\ell\, \lambda^\ell + \cdots + A_1\, \lambda + A_0 \in \mathbb{F}[\lambda]^{n\times n}. \tag{2.14}$$

System (2.13) has at least one polynomial solution in t *of degree* p *such that*

$$\begin{cases} p = d & \text{if} \quad 0 \notin \sigma(L), \\ d \le p \le d + \max\{\kappa_j, j = 1, \ldots, m_g(0)\} & \text{if} \quad 0 \in \sigma(L), \end{cases}$$

where $\kappa_1, \ldots, \kappa_{m_g(0)}$ *denote the partial multiplicities of* $L(\lambda)$ *at the eigenvalue* 0.

Proof. We give here a proof using the associated non-homogeneous differential system of first-order

$$\vartheta(Y) = \mathcal{C}\,Y + \Phi(t), \tag{2.15}$$

where $\mathcal{C}$ is the block companion matrix (2.3) with the $\tilde{A}_i = -A_\ell^{-1}A_i$ constant and $\Phi(t) = \begin{pmatrix} 0 & \cdots & 0 & \phi(t)^T \end{pmatrix}^T$.

Write $Y = \sum_{i=0}^{p} Y_i \frac{t^i}{i!}$ with $Y_i \in \mathbb{F}^{n\ell}$ and $Y_p \neq 0$ and $\Phi(t) = \sum_{i=0}^{d} \Phi_i \frac{t^i}{i!}$ with $\Phi_i \in \mathbb{F}^{n\ell}$ and $\Phi_d \neq 0$. Since $\vartheta(\log^i(x)) = \frac{d}{dt}(t^i) = it^{i-1}$, the degree in t of $\vartheta(Y) - \mathcal{C}\,Y$ is less than or equal to that of Y. Hence we have $d \leq p$. Suppose first that $\mathcal{C}$ is invertible, *i.e.*, 0 is not an eigenvalue of $L(\lambda)$ ($I\lambda - \mathcal{C}$ is a linearization of the matrix polynomial $L(\lambda)$, see Section 1.3 of Chapter 1). Plugging $Y = \sum_{i=0}^{p} Y_i \frac{t^i}{i!}$ into (2.15), we find that the Y_i must satisfy

$$\begin{cases} \mathcal{C}\,Y_p = -\Phi_p, \\ \mathcal{C}\,Y_i = Y_{i+1} - \Phi_i, \quad i = p-1, \ldots, 0, \end{cases}$$

with $\Phi_i = 0$ for $i > d$. Since $\mathcal{C}$ is supposed to be invertible, we necessarily have $p = d$ and the Y_i for $i = p, \ldots, 0$ can be computed recursively which provides a polynomial solution of (2.15) of degree $p = d$. Suppose now that $\mathcal{C}$ is not invertible. Let P be an invertible constant matrix of size $n\,\ell$ such that the change of variables $Z = P\,Y$ transforms System (2.15) into $\vartheta(Z) = \mathcal{J}Z + \Psi(t)$, where $\mathcal{J}$ is the Jordan normal form of $\mathcal{C}$ and $\Psi(t) = P\,\Phi(t)$. Note that $\Psi(t)$ is a vector polynomial in t of degree d since P is invertible. We can suppose without any loss of generality that $\mathcal{J} = \operatorname{diag}(\mathcal{J}_1, \mathcal{J}_2)$ with $\mathcal{J}_1$ nilpotent and $\mathcal{J}_2$ invertible. Decompose Z and $\Psi(t)$ into blocks according to the decomposition of $\mathcal{J}$, *i.e.*,

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \quad \text{and} \quad \Psi(t) = \begin{pmatrix} \Psi_1(t) \\ \Psi_2(t) \end{pmatrix}.$$

System (2.15) is then equivalent to

$$\vartheta(Z_i) = \mathcal{J}_i\,Z_i + \Psi_i(t), \quad \text{with } i = 1, 2. \tag{2.16}$$

Since $\mathcal{J}_2$ is invertible, System (2.16) for $i = 2$ has a polynomial solution in t of degree equal to that of $\Psi_i(t)$, namely, bounded by d. Now for $i = 1$, the problem can be reduced to that of finding polynomial solutions of the first-order differential system of size κ

$$\vartheta(X) = JX + F(t), \tag{2.17}$$

where κ is a partial multiplicity of $L(\lambda)$ associated with eigenvalue 0, J is a nilpotent Jordan block of the form

$$J = \begin{pmatrix} 0 & 1 & & (0) \\ & \ddots & \ddots & \\ & & & 1 \\ & (0) & & 0 \end{pmatrix}$$

and $F(t)$ is a vector polynomial in t of degree bounded by d. Let $X = \begin{pmatrix} X_1 & \cdots & X_\kappa \end{pmatrix}^T$ and $F(t) = \begin{pmatrix} F_1(t) & \cdots & F_\kappa(t) \end{pmatrix}^T$. System (2.17) can then be written as

$$\begin{cases} \vartheta(X_\kappa) = F_\kappa(t), \\ \vartheta(X_i) = X_{i+1} + F_i(t), \quad \text{for } i = \kappa - 1, \ldots, 1. \end{cases}$$

Since $F_\kappa(t)$ is a polynomial in t, X_κ can be obtained by integration and is also a polynomial in t with $\deg(X_\kappa) = \deg(F_\kappa) + 1$. Proceeding recursively, we can thus compute the polynomials X_i for $i = \kappa - 1, \ldots, 1$. The degree in t of the polynomial solution X of (2.17) that we obtain is then less than or equal to $\kappa + d$. □

The general polynomial solution in $\log(x)$ of System (2.13) follows directly from the above proposition and from the fact that a polynomial solution in $t = \log(x)$ of the homogeneous system associated with (2.13) is a regular solution of exponent 0 (if of course $L(\lambda)$ given in (2.10) has 0 as an eigenvalue).

Corollary 2.5.1. *Consider a non-homogeneous linear differential system with constant coefficients of the form (2.13) and let $z(x)$ be a particular polynomial solution in $\log(x)$ of the system. If $0 \notin \sigma(L)$ where $L(\lambda)$ is the matrix polynomial given in (2.14), then $z(x)$ is the unique polynomial solution in $\log(x)$ of System (2.13). Otherwise, let $v_{i,0}, \cdots, v_{i,\kappa_i-1}$ for $i = 1, \ldots, m_g(0)$ be a canonical set of Jordan chains for $L(\lambda)$ associated with eigenvalue 0. For $i = 1, \ldots, m_g(0)$ and $j = 0, \ldots, \kappa_i - 1$, let*

$$y_{i,j}(x) = v_{i,j} + v_{i,j-1}\frac{\log(x)}{1!} + \cdots + v_{i,0}\frac{\log^j(x)}{j!}.$$

Then, the general polynomial solution in $\log(x)$ of (2.13) is given by

$$y(x) = \sum_{\substack{1 \le i \le m_g(0) \\ 0 \le j \le \kappa_i - 1}} K_{i,j}\, y_{i,j}(x) + z(x),$$

where the $K_{i,j}$ are arbitrary constants in $\mathbb{F}$.

2.5.2 Computation of the general polynomial solution in $t = \log(x)$

We will now explain how to compute efficiently the general polynomial solution in $t = \log(x)$ of systems of the form (2.13). Write the right-hand side $\phi(t)$ of (2.13) under the form $\phi(t) = \sum_{i=0}^{d} q_i \frac{t^i}{i!}$, where $q_i \in \mathbb{F}^n$ and $q_d \neq 0$. Write the general polynomial solution of (2.13) in the form $y = \sum_{i=0}^{p} y_i \frac{t^i}{i!}$, where $y_i \in \mathbb{F}^n$ are to be determined and

$$\begin{cases} p = d & \text{if} \quad 0 \notin \sigma(L), \\ p = d + \max\{\kappa_j, j = 1, \ldots, m_g(0)\} & \text{if} \quad 0 \in \sigma(L). \end{cases} \tag{2.18}$$

Writing $L(\vartheta) = \sum_{i=0}^{\ell} \frac{1}{i!} L^{(i)}(0)\, \vartheta^i$, we see that solving System (2.13) is equivalent to solving the linear system of size $(p+1)\,n$

$$\underbrace{\begin{pmatrix} L(0) & & & & \\ \vdots & \ddots & & (0) & \\ \frac{1}{(p-d)!}L^{(p-d)}(0) & & \ddots & & \\ \vdots & & & \ddots & \\ \frac{1}{p!}L^{(p)}(0) & \cdots & \cdots & \cdots & L(0) \end{pmatrix}}_{T_{p+1}} \begin{pmatrix} y_p \\ \vdots \\ y_d \\ \vdots \\ y_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ q_d \\ \vdots \\ q_0 \end{pmatrix}, \tag{2.19}$$

which is equivalent to solving the $p+1$ linear systems of size n

$$L(0)\, y_i = f_i, \quad \text{for } i = p, \ldots, 0, \tag{2.20}$$

where $f_p = q_p$, $f_i = q_i - \sum_{j=1}^{p-i} \frac{1}{j!} L^{(j)}(0)\, y_{i+j}$ for $i = p-1, \ldots, 0$, and $q_i = 0$ for $i > d$. Note that all the systems in (2.20) share the same matrix $L(0)$.

If 0 is not an eigenvalue of $L(\lambda)$, then the y_i for $i = p, \ldots, 0$ are given by $y_i = L^{-1}(0)\, f_i$. Otherwise, compute an LU decomposition of $L(0)$ and then use it to compute recursively the y_i's

in the following way. Let $L(0) = \mathbf{P}\,\mathbf{L}\,\mathbf{U}$ where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ a lower triangular matrix having 1 on its diagonal and $\mathbf{U}$ an upper triangular matrix whose last $n - \text{rank}(L(0)) = m_g(0)$ rows are zero. Since the κ_i's are positive integers, the integer p is then greater than d and hence y_p satisfies the system $L(0)\,y_p = 0$. Thus, computing y_p is equivalent to solving the system $\mathbf{U}\,y_p = 0$. It follows that the last $m_g(0)$ components of y_p are arbitrary constants. Suppose now that we have already computed the y_j's for $j = i+1, \ldots, p$. These vectors depend on arbitrary constants since $L(0)$ is a singular matrix. Let us now explain how to solve the system $L(0)\,y_i = f_i$. We first solve the system $\mathbf{L}\,z = \mathbf{P}^T f_i$. Since $\mathbf{L}$ is invertible, the solution z depends only on the arbitrary constants appearing in f_i. Then, we consider the system $\mathbf{U}\,y_i = z$. As it has to be consistent, the values of some arbitrary constants of z (and hence some arbitrary constants of y_j for $j = i+1, \ldots, p$) are fixed so that the last $m_g(0)$ components of z can be zero. Thus, the system can be solved and one gets y_i and so on.

We summarize this in the following algorithm.

ALGORITHM **GenPolSol_NHS**

INPUT: The matrices $L(0), L'(0), \ldots, L^{(\ell)}(0)$, the right-hand side $\phi(t)$ of System (2.13), the matrix $L^{-1}(0)$ if $0 \notin \sigma(L)$, and the partial multiplicities $\kappa_1, \ldots, \kappa_{m_g(0)}$ at 0, otherwise.
OUTPUT: The general polynomial solution in $t = \log(x)$ of System (2.13).

INITIALIZATION: Write $\phi(t)$ under the form $\phi(t) = \sum_{i=0}^{d} q_i \frac{t^i}{i!}$ with $q_i \in \mathbb{F}^n$ and $q_d \neq 0$;

1. **If** 0 is not an eigenvalue of $L(0)$ **then**
   1.1. Let $p = d$;
   1.2. **For** i from p by -1 to 0 **do**
      1.2.(a). **If** $i < p$ **then** compute $f_i = q_i - \sum_{j=1}^{p-i} \frac{1}{j!} L^{(j)}(0)\,y_{i+j}$ **else** $f_i = q_i$ **end if**;
      1.2.(b). Compute $y_i = L^{-1}(0)\,f_i$;
      end do;
2. **else**
   2.1. Let $p = d + \max\{\kappa_i, i = 1, \ldots, m_g(0)\}$, $r = n - m_g(0)$ and $q_i = 0$ for $i > d$;
   2.2. Compute an LU decomposition of $L(0)$, namely, $L(0) = \mathbf{P}\,\mathbf{L}\,\mathbf{U}$;
   2.3. **For** i from p by -1 to 0 **do**
      2.3.(a). **If** $i < p$ **then** compute $f_i = q_i - \sum_{j=1}^{p-i} \frac{1}{j!} L^{(j)}(0)\,y_{i+j}$ **else** $f_i = q_i$ **end if**;
      2.3.(b). Solve the linear system $\mathbf{L}\,z = \mathbf{P}^T f_i$;
      2.3.(c). Solve the linear system $\{z_j = 0, \text{ for } j = r+1, \ldots, n\}$;
         // Here z_j denotes the jth component of vector z
      2.3.(d). Update z and y_j for $j = i+1, \ldots, p$;
      2.3.(e). Compute the general solution of system $\mathbf{U}\,y_i = z$;
      end do;
   end if;
3. **Return** $\sum_{i=0}^{p} y_i \frac{t^i}{i!}$;

Proposition 2.5.2. *Algorithm* **GenPolSol_NHS** *is correct,* i.e.*, the number of arbitrary constants in its output is equal to the dimension of* $\ker(T_{p+1})$*, where* T_{p+1} *is the matrix defining System (2.19).*

Proof. When $0 \notin \sigma(L)$, this is obvious since there exists a unique solution and matrix T_{p+1} is invertible. Hence, we consider the case where $0 \in \sigma(L)$. For sake of simplicity, we will suppose that the right-hand side $\phi(t)$ is equal to zero since the arbitrary constants in the output come

from the solution of the homogeneous system associated with (2.19). More generally, we will show that, for $i = i_0$ $(0 \le i_0 \le p)$ in the **For** loop, the vector

$$\begin{pmatrix} y_p^{(i_0)} \\ \vdots \\ y_{i_0}^{(i_0)} \end{pmatrix},$$

where $y_p^{(i_0)}, \ldots, y_{i_0}^{(i_0)}$ denote the vectors $y_p, \ldots, y_{i_0}$ just after executing Step 2.3.(e), is the general solution of system $T_{p-i_0+1}\, X = 0$, where

$$T_{p-i_0+1} = \begin{pmatrix} L(0) & & (0) \\ \vdots & \ddots & \\ \frac{1}{(p-i_0)!} L^{(p-i_0)}(0) & \cdots & L(0) \end{pmatrix}.$$

In other words, we will show that the number of arbitrary constants in $y_p^{(i_0)}, \ldots, y_{i_0}^{(i_0)}$ is equal to the dimension of $\ker(T_{p-i_0+1})$. We proceed by induction. First, for $i = p$, we have $f_p = 0$ and the vector $y_p^{(p)}$, as computed in Step 2.3.(e), is the general solution of System $\mathbf{U}X = 0$ and therefore it is so for the system $T_1\, X = 0$ with $T_1 = L(0)$. Hence, it is satisfied for $i = p$. Suppose now that it is satisfied for $i = i_0 + 1$, that is, the number of arbitrary constants in $y_p^{(i_0+1)}, \ldots, y_{i_0+1}^{(i_0+1)}$ is equal to the dimension of $\ker(T_{p-i_0})$, and let us show it for $i = i_0$. Write

$$\begin{pmatrix} y_p^{(i_0+1)} \\ \vdots \\ y_{i_0+1}^{(i_0+1)} \end{pmatrix} = V_{i_0+1}\, C_{i_0+1},$$

where C_{i_0+1} is the vector of the arbitrary constants appearing in $y_p^{(i_0+1)}, \ldots, y_{i_0+1}^{(i_0+1)}$ and V_{i_0+1} is a matrix whose columns form a basis of the matrix T_{p-i_0}. In Step 2.3.(a), f_{i_0} can be written as

$$f_{i_0} = -T_{p-i_0}^*\, V_{i_0+1}\, C_{i_0+1} \quad \text{with } T_{p-i_0}^* = \begin{pmatrix} \frac{1}{(p-i_0)!} L^{(p-i_0)}(0) & \cdots & L'(0) \end{pmatrix}.$$

Hence, z in Step 2.3.(b) is given by $z = -\mathbf{L}^{-1}\, \mathbf{P}^T\, T_{p-i_0}^*\, V_{i_0+1}\, C_{i_0+1}$. Let s_{i_0} denote the rank of the system in Step 2.3.(c); s_{i_0} is indeed equal to the rank of the submatrix of $\mathbf{L}^{-1}\, \mathbf{P}^T\, T_{p-i_0}^*\, V_{i_0+1}$ composed of rows of indices from $r+1$ to n, where $r = \operatorname{rank}(L(0)) = n - m_g(0)$. However, we have seen in Chapter 1 (see Equation (1.13)) that

$$\operatorname{rank}\begin{pmatrix} T_{p-i_0}^*\, V_{i_0+1} & L(0) \end{pmatrix} = \operatorname{rank}(T_{p-i_0+1}) - \operatorname{rank}(T_{p-i_0}).$$

Thus, by multiplying on the left by the invertible matrix $\mathbf{L}^{-1}\, \mathbf{P}^T$, we find

$$\operatorname{rank}\begin{pmatrix} \mathbf{L}^{-1}\, \mathbf{P}^T\, T_{p-i_0}^*\, V_{i_0+1} & U \end{pmatrix} = \operatorname{rank}(T_{p-i_0+1}) - \operatorname{rank}(T_{p-i_0}).$$

From the particular structure of U and the equality just above, it follows that

$$s_{i_0} + r = \operatorname{rank}(T_{p-i_0+1}) - \operatorname{rank}(T_{p-i_0}).$$

In Step 2.3.(d), the values of s_{i_0} arbitrary constants are determined and in Step 2.3.(e), $m_g(0)$ new arbitrary constants are introduced. Hence, the number of arbitrary constants in $y_p^{(i_0)}, \ldots, y_{i_0}^{(i_0)}$ is equal to

$$\begin{aligned} \dim(\ker(T_{p-i_0})) - s_{i_0} + m_g(0) &= \underbrace{\dim(\ker(T_{p-i_0})) - \operatorname{rank}(T_{p-i_0+1})}_{=\, n\,(p-i_0)} + \operatorname{rank}(T_{p-i_0}) + \underbrace{r + m_g(0)}_{=\, n} \\ &= n\,(p - i_0 + 1) - \operatorname{rank}(T_{p-i_0+1}) \\ &= \dim(\ker(T_{p-i_0+1})) \qquad \textit{QED.} \end{aligned}$$

Thus, for $i_0 = 0$, the number of arbitrary constants in $y_p^{(0)}, \ldots, y_0^{(0)}$, which appear in the output of the algorithm, is equal to the dimension of $\ker(T_{p+1})$, which ends the proof. □

Remark 2.5.1. From the proof of Proposition 2.5.2 and Lemma 1.6.1 of Chapter 1, we can deduce that the number of arbitrary constants appearing in the $y_i's$ during all steps of Algorithm **GenPolSol_NHS** never exceeds $m_a(0)$.

Proposition 2.5.3. *Algorithm* **GenPolSol_NHS** *returns the general polynomial solution of System (2.13) after at most* $O\big(n^2\, p\, m_a(0)\, \min\{p, \ell\} + n\, p^2\, m_g(0)\, m_a(0) + n^3\big)$ *operations in* $\mathbb{F}$ *if* $0 \in \sigma(L)$, *and* $O\big(n^2\, p\, \min\{p, \ell\}\big)$ *operations in* $\mathbb{F}$, *otherwise. Here,* p *is given by (2.18).*

Proof. Step 1.2.(a) requires $O\big(n^2 \min\{p, \ell\}\big)$ operations in $\mathbb{F}$ while Step 1.2.(b) can de done in $O\big(n^2\big)$ operations in $\mathbb{F}$. As Step 1.2 is repeated $p+1$ times, Step 1 costs at most $O\big(n^2\, p \min\{p, \ell\}\big)$ operations in $\mathbb{F}$.

Let us now determine the cost of Step 2. Step 2.2 can be done using at most $O\big(n^3\big)$ operations in $\mathbb{F}$. From Remark 2.5.1, it follows that Step 2.3.(a) can be done in at most $O\big(n^2\, m_a(0) \min\{p, \ell\}\big)$ operations in $\mathbb{F}$. Since $\mathbf{L}$ and $\mathbf{U}$ are triangular matrices and since vectors f_i and z depend on at most $m_a(0)$ arbitrary constants, Steps 2.3.(b) and 2.3.(e) can be done using at most $O\big(n^2\, m_a(0)\big)$ operations in $\mathbb{F}$. Let us now determine the cost of Step 2.3.(c). The components of z are linear combinations of the arbitrary constants appearing in y_j for $j = i+1, \ldots, p$ and hence solving the system in Step 2.3.(c) can be done in at most $O\big((n-r)^2\, m_a(0)\big) = O\big(m_g(0)^2\, m_a(0)\big)$ operations in $\mathbb{F}$. It remains to compute the cost of Step 2.3.(d). Let $s \leq m_g(0)$ denote the rank of the system in Step 2.3.(c). Solving the system in Step 2.3.(c) determines the values of s arbitrary constants of z as linear combinations of the other arbitrary constants whose number is bounded by $m_a(0) - s$. Updating one component of z costs then at most $O(s\,(m_a(0) - s)) = O(m_g(0)\, m_a(0))$ operations in $\mathbb{F}$. Since the number of components to be updated in vectors z and y_j for $j = i+1, \ldots, p$ is bounded by $n\,(p-i+1)$, Step 2.3.(d) can be done in at most $O(n\, p\, m_g(0)\, m_a(0))$ operations in $\mathbb{F}$. Thus, one passage through the **For** loop costs at most $O\big(n^2\, m_a(0) \min\{p, \ell\} + n\, p\, m_g(0)\, m_a(0)\big)$ operations in $\mathbb{F}$. Hence, Step 2.3 costs $O\big(n^2\, p\, m_a(0)\, \min\{p, \ell\} + n\, p^2\, m_g(0)\, m_a(0)\big)$ operations in $\mathbb{F}$. Step 2 can then be done using at most $O\big(n^2\, p\, m_a(0)\, \min\{p, \ell\} + n\, p^2\, m_g(0)\, m_a(0) + n^3\big)$ operations in $\mathbb{F}$. □

2.6 A direct method for computing regular solutions

In this section, we develop a new approach for computing a basis of the regular formal solution space of systems of the form (2.1). This approach is inspired by Poole's method [86] for computing regular solutions of scalar linear differential equations.

2.6.1 Description of the approach

The idea behind Poole's method consists in looking for regular solutions viewed as formal power series whose coefficients are vectors of polynomials in $t = \log(x)$, *i.e.*, of the form

$$y(x) = \sum_{m=0}^{\infty} U_m(t)\, x^{\lambda_0 + m}, \tag{2.21}$$

where $\lambda_0 \in \overline{\mathbb{K}}$, for all $m \geq 0$, $U_m(t) \in \overline{\mathbb{K}}[t]^n$ and $U_0 \neq 0$. We refer to $y(x)$ in (2.21) as a *regular solution of exponent* λ_0.

Write the coefficient matrices $A_i(x)$ of System (2.1) as

$$A_i(x) = \sum_{j=0}^{\infty} A_{i,j}\, x^j, \tag{2.22}$$

where $A_{i,j} \in \mathbb{K}^{n\times n}$. With System (2.1), we associate the matrix polynomials defined by

$$\forall\, i \geq 0, \quad L_i(\lambda) = A_{\ell,i}\,\lambda^{\ell} + \cdots + A_{1,i}\,\lambda + A_{0,i}. \tag{2.23}$$

Therefore, System (2.1) can be written in the form

$$\mathcal{L}(x,\vartheta)(y(x)) = \sum_{i=0}^{\infty} x^i\, L_i(\vartheta)(y(x)) = 0. \tag{2.24}$$

Substituting (2.21) into (2.24), we find

$$\sum_{i=0}^{\infty} x^i\, L_i(\vartheta)\left(\sum_{m=0}^{\infty} U_m\, x^{\lambda_0+m}\right) = \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} x^i\, L_i(\vartheta)\left(U_m\, x^{\lambda_0+m}\right) = 0.$$

Identifying the coefficients of the powers of x to 0, we find that λ_0 and U_0 must satisfy

$$L_0(\vartheta)\left(x^{\lambda_0}\, U_0\right) = 0, \tag{2.25}$$

and then using the equality

$$L_i(\vartheta)\left(x^{\lambda_0+m}\, U_m\right) = x^{\lambda_0+m}\, L_i(\vartheta+\lambda_0+m)\,(U_m)\,,$$

we find that, for $m \geq 1$, U_m satisfies

$$L_0(\vartheta+\lambda_0+m)\,(U_m) = -\sum_{i=0}^{m-1} L_{m-i}(\vartheta+\lambda_0+i)\,(U_i)\,. \tag{2.26}$$

The problem is then reduced to finding λ_0 and $U_0, U_1, \ldots \in \overline{\mathbb{K}}[t]^n$ satisfying (2.25) and (2.26).

Equation (2.25) shows that $x^{\lambda_0}\, U_0$ is a regular solution of the Euler matrix differential equation defined by $L_0(\vartheta)(y(x)) = 0$. Hence, by Lemma 2.4.1, λ_0 must be chosen as an eigenvalue of $L_0(\lambda)$ and U_0 of the form

$$U_0 = v_{k-1} + v_{k-2}\,\frac{\log(x)}{1!} + \cdots + v_0\,\frac{\log^{k-1}(x)}{(k-1)!}, \tag{2.27}$$

where $k \in \mathbb{N}^*$ and $v_0, \ldots, v_{k-1}$ form a Jordan chain of length k for $L_0(\lambda)$ associated with λ_0. We recall that k is less than or equal to one of the partial multiplicities of $L_0(\lambda)$ at λ_0.

Let $\lambda_0 \in \sigma(L_0)$ and U_0 the vector polynomial in $t = \log(x)$ of degree $k-1$ given by (2.27). Consider first Equation (2.26) for $m = 1$. This is a non-homogeneous linear differential system with constant coefficients in $\mathbb{K}(\lambda_0)$ and polynomial right-hand side in $t = \log(x)$ of degree less than k. According to Proposition 2.5.1, this system admits at least one polynomial solution U_1 in t. The degree of this latter is bounded by k if $\lambda_0 + 1$ is not an eigenvalue of $L_0(\lambda)$ (which is equivalent to saying that 0 is not an eigenvalue of $L_0(\lambda+\lambda_0+1)$) and by $k + \max_j\{\kappa_j(\lambda_0+1)\}$, otherwise, where the $\kappa_j(\lambda_0+1)$ denote the partial multiplicities of $L_0(\lambda)$ at the eigenvalue λ_0+1. Recursively, we can then obtain $U_2, U_3, \ldots \in \mathbb{K}(\lambda_0)[t]^n$ as polynomial solutions in $t = \log(x)$ of the non-homogeneous systems (2.26) for $m = 2, 3, \ldots$. From Proposition 2.5.1, it follows that the degree in t of U_m is bounded by

$$k + \sum_{1\leq i\leq m \text{ s.t. } \lambda_0+i\in\sigma(L_0)} \max_j\{\kappa_j(\lambda_0+i)\}\,, \tag{2.28}$$

where the $\kappa_j(\lambda_0+i)$ denote the partial multiplicities of $L_0(\lambda)$ at the eigenvalue λ_0+i.

In the sequel, we propose two versions of this approach. In the first version, we propose to compute a basis of the regular formal solution space of System (2.1) starting from a basis of the regular formal solution space of System (2.25) that we obtain using Algorithm **Euler_MDE**. In the second version, we propose to split the spectrum of $L_0(\lambda)$ into pairwise disjoint subsets $\sigma_1, \ldots, \sigma_r$ such that two eigenvalues belonging to two different subsets σ_i and σ_j do not differ by integers[2], then we compute the general regular solution associated with each subset σ_i. In the latter version, we do not use the notion of partial multiplicities nor that of Jordan chains.

2.6.2 First version: using Jordan chains

We have shown earlier that every regular solution $x^{\lambda_0} U_0$ of System (2.25) can be extended to construct a regular formal solution of System (2.1) of exponent λ_0. Thus, starting from a basis of the regular formal solution space of System (2.25), one can construct $n\,\ell$ regular solutions of System (2.1). Furthermore, these solutions are linearly independent since their first terms $x^{\lambda_0} U_0$ are so. Therefore, we obtain a basis of the regular formal solution space of System (2.1).

Algorithm **BCE_V1** below takes as input the coefficients $A_i(x)$ of System (2.1) and an integer $\nu \in \mathbb{N}$ and returns a basis of the regular formal solution space of System (2.1) where the series involved are computed up to order ν, that is, only the coefficients of x^i with $i \leq \nu$ are computed.

Remark 2.6.1. From Equation (2.26), we can deduce that the computation of a regular solution up to order ν of System (2.1) requires only the knowledge of the operators $L_0(\vartheta), L_1(\vartheta), \ldots, L_\nu(\vartheta)$ given by (2.23). These operators depend only on the first $\nu+1$ coefficients of the expansion (2.22). Consequently, it suffices to truncate the entries of the coefficient matrices $A_i(x)$ of System (2.1) at order ν, *i.e.*, $A_i(x) = \sum_{j=0}^{\nu} A_{i,j}\, x^j + O\big(x^{\nu+1}\big)$.

ALGORITHM **BCE_V1**

INPUT: An integer $\nu \in \mathbb{N}$ and the coefficient matrices $A_i(x)$ of System (2.1) truncated at order ν.
OUTPUT: A basis of the regular formal solution space of System (2.1) where the series involved are truncated at order ν.

INITIALIZATION: Let $L_0(\lambda) = \sum_{i=0}^{\ell} A_i(0)\,\lambda^i$ and Sol = { };

1. Compute $\sigma(L_0)$;
2. **For** each element λ_0 of $\sigma(L_0)$ **do**
   2.1. Compute a canonical set of Jordan chains $v_{i,0}, \ldots, v_{i,\kappa_i(\lambda_0)-1}$ for $i = 1, \ldots, m_g(\lambda_0)$ for $L_0(\lambda)$ associated with λ_0;
   2.2. **For** i from 1 to $m_g(\lambda_0)$ **do**
      For j from 0 to $\kappa_i(\lambda_0) - 1$ **do**
         Let $U_{(i,j),0} = \sum_{k=0}^{j} v_{i,j-k} \frac{\log^k(x)}{k!}$;
         For m from 1 to ν **do**
            Compute $U_{(i,j),m}$ as solution of System
            $L_0(\vartheta + \lambda_0 + m)\,\big(U_{(i,j),m}\big) = -\sum_{k=0}^{m-1} L_{m-k}(\vartheta + \lambda_0 + k)\,\big(U_{(i,j),k}\big)$;
         end do;
         Let Sol = Sol $\cup \big\{\sum_{m=0}^{\nu} U_{(i,j),m}\, x^{\lambda_0+m}\big\}$;
      end do;

[2]This implies that the eigenvalues belonging to the same subset σ_i differ by nonzero integers.

```
        end do;
  end do;
3. Return Sol;
```

Remark 2.6.2. Let us make some comments concerning our MAPLE implementation of Algorithm **BCE_V1** above.

1. To compute $U_{(i,j),m}$ (for $m = 1, \ldots, \nu$) in Step 2.2, we need to check whether $\lambda_0 + m$ with $\lambda_0 \in \sigma(L_0)$ is an element of $\sigma(L_0)$ or not. This leads us to consider the problem in a more general form: check whether two distinct elements of $\sigma(L_0)$ (see Remark 1.1.1 of Chapter 1) differ or not by nonzero integer. To achieve this in practice, we rely on two facts. The first fact is that the roots of an irreducible polynomial over $\mathbb{K}[\lambda]$ do not differ by integers. The second fact is that if $p_1(\lambda)$ and $p_2(\lambda)$ are two irreducible polynomials over $\mathbb{K}[\lambda]$ and if there exist a root α of $p_1(\lambda)$ and a nonzero integer s such that $\alpha + s$ is a root of $p_2(\lambda)$, that is,
$$p_1(\alpha) = p_2(\alpha + s) = 0,$$
then $p_1(\lambda) = p_2(\lambda + s)$. Thus, the roots of two monic irreducible polynomials $p_1(\lambda)$ and $p_2(\lambda)$ of same degree differ by a nonzero integer if and only if s defined by
$$s := \frac{\mathrm{tr}(p_1(\lambda)) - \mathrm{tr}(p_2(\lambda))}{\deg(p_1(\lambda))},$$
where $\mathrm{tr}(p_i(\lambda))$ denote the trace of the polynomial $p_i(\lambda)$ for $i = 1, 2$, is a nonzero integer and $p_1(\lambda) = p_2(\lambda + s)$.

2. For each $\lambda_0 \in \sigma(L)$, $i \in \{1, \ldots, m_g(\lambda_0)\}$ and $m \in \{1, \ldots, \nu\}$, we only compute $U_{(i,\kappa_i(\lambda_0)-1),m}$ (*i.e.*, for $j = \kappa_i(\lambda_0) - 1$) since from it we can deduce $U_{(i,j),m}$ for $j = 0, \ldots, \kappa_i(\lambda_0) - 2$. Indeed, let p_m denote the degree of $U_{(i,\kappa_i(\lambda_0)-1),m}$ and write $U_{(i,\kappa_i(\lambda_0)-1),m}$ as a vector polynomial in t as follows
$$U_{(i,\kappa_i(\lambda_0)-1),m} = \sum_{s=0}^{p_m} U_{(i,\kappa_i(\lambda_0)-1),(m,s)} \frac{t^s}{s!},$$
where $U_{(i,\kappa_i(\lambda_0)-1),(m,s)} \in \overline{\mathbb{K}}^n$ for $s = 0, \ldots, p_m$. For $j = 0, \ldots, \kappa_i(\lambda_0) - 2$, we take
$$U_{(i,j),m} = \sum_{s=0}^{|p_m - \kappa_i(\lambda_0)+1|+j} U_{(i,\kappa_i(\lambda_0)-1),(m,\kappa_i(\lambda_0)-1-j+s)} \frac{t^s}{s!},$$
with $U_{(i,\kappa_i(\lambda_0)-1),(m,s)} = 0$ if $s > p_m$.

3. We compute the vector polynomial $U_{(i,\kappa_i(\lambda_0)-1),m}$ using Algorithm **GenPolSol_NHS** by giving as input the matrices $L_0(\lambda_0 + m), L_0'(\lambda_0 + m), \ldots, L_0^{(\ell)}(\lambda_0 + m)$, the right-hand side of the system satisfied by $U_{(i,\kappa_i(\lambda_0)-1),m}$, the matrix $L_0^{-1}(\lambda_0 + m)$ if $\lambda_0 + m \notin \sigma(L_0)$, and the partial multiplicities of $L_0(\lambda)$ at $\lambda_0 + m$, otherwise. The algorithm will return the general polynomial solution, *i.e.*, when $\lambda_0 + m \in \sigma(L_0)$, the output of Algorithm **GenPolSol_NHS** will depend on arbitrary constants. Since we need any solution $U_{(i,\kappa_i(\lambda_0)-1),m}$, we replace then these arbitrary constants by zero.

4. If the eigenvalues of $L_0(\lambda)$ differ by integers, then we proceed with them in Step 2 by decreasing order. Hence the partial multiplicities of $L_0(\lambda)$ at $\lambda_0 + m$ (if $\lambda_0 + m \in \sigma(L_0)$) needed to compute $U_{(i,\kappa_i(\lambda_0)-1),m}$ are already known without any extra computation.

Proposition 2.6.1. *Given a system of n linear differential equations of order ℓ of the form (2.1), Algorithm* **BCE_V1** *returns a basis of its regular formal solution space using at most $O\big(n^4\,\ell^3\,\nu^2+n^6\,\ell^4\big)$ operations in $\mathbb{K}$ where ν denotes the order of truncation of the series.*

Proof. We first compute the matrix inverse $L_0^{-1}(\lambda)$ and the first ℓ derivatives of the matrices $L_i(\lambda)$ for $i=0,\ldots,\nu$ as it is needed to compute the coefficients $U_{(i,j),m}$. Computing the inverse of an $n\times n$ matrix polynomial of degree ℓ can be done using $O^{\sim}\big(n^3\,\ell\big)$ operations in $\mathbb{K}$ (see [61]). Computing the derivatives reduces to differentiate $n^2(\nu+1)$ polynomials of degree less than or equal to ℓ at most ℓ times. Each differentiation costs $O(\ell)$ operations in $\mathbb{K}$. Hence, the cost of computing the $L_i^{(j)}(\lambda)$ for $i=0,\ldots,\nu$ and $j=1,\ldots,\ell$ is bounded by $O\big(n^2\,\ell^2\,\nu\big)$ operations in $\mathbb{K}$.

For each element of $\sigma(L_0)$, using Algorithm **CanonicalSet_JC**, Step 2.1 can be performed in at most $O\big(n^4\,\ell^3\,m_a(\lambda_0)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$ (see Remark 1.6.2 of Chapter 1). Let us now determine the cost of computing one $U_{(i,\kappa_i-1),m}$[3] (see Remark 2.6.2). For this, we need first to determine the cost of computing the right-hand side $-\sum_{k=0}^{m-1} L_{m-k}(\vartheta+\lambda_0+k)\,\big(U_{(i,\kappa_i-1),k}\big)$ of the system satisfied by $U_{(i,\kappa_i-1),m}$. Let us study the cost of applying $L_{m-k}(\vartheta+\lambda_0+k)$ to $U_{(i,\kappa_i-1),k}$. Since the degree of $U_{(i,\kappa_i-1),k}$ in $t=\log(x)$ is less than $n\,\ell$, write $U_{(i,\kappa_i-1),k}=\sum_{j=0}^{n\ell-1} U_{(i,\kappa_i-1),(k,j)}\,\frac{t^j}{j!}$ with $U_{(i,\kappa_i-1),(k,j)}\in\mathbb{K}(\lambda_0)^n$. Computing $L_{m-k}(\vartheta+\lambda_0+k)\,\big(U_{(i,\kappa_i-1),k}\big)$ is then equivalent to performing the matrix-vector product

$$\begin{pmatrix} L_{m-k}(\lambda_0+k) & & (0) \\ \vdots & \ddots & \\ \frac{1}{(n\,\ell-1)!}L_{m-k}^{(n\ell-1)}(\lambda_0+k) & \cdots & L_{m-k}(\lambda_0+k) \end{pmatrix} \begin{pmatrix} U_{(i,\kappa_i-1),(k,n\ell-1)} \\ \vdots \\ U_{(i,\kappa_i-1),(k,0)} \end{pmatrix}. \tag{2.29}$$

To achieve this multiplication, we have to evaluate $L_{m-k}^{(j)}(\lambda)$ for $j=0,\ldots,\ell$ at $\lambda=\lambda_0+k$ ($L_{m-k}^{(j)}(\lambda)=0$ for $j>\ell$). This is equivalent to evaluate at most $n^2\,(\ell+1)$ polynomials of degree bounded by ℓ at λ_0+k. Evaluating a polynomial of degree less than or equal to ℓ with coefficients in $\mathbb{K}$ at $\lambda_0+k\in\mathbb{K}(\lambda_0)$ can be done using at most $O(\ell\,d_{\lambda_0})$ operations in $\mathbb{K}$. Hence, the cost of the evaluation is $O\big(n^2\,\ell^2\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Now, the multiplication in (2.29) requires $n\,\ell^2+n\,\ell-\frac{\ell}{2}\,(\ell+1)$ products of an $n\times n$ matrix by an n-dimensional vector with entries in $\mathbb{K}(\lambda_0)$. Consequently, the multiplication in (2.29) costs at most $O\big(n^3\,\ell^2\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Thus, computing the right-hand side of the system satisfied by $U_{(i,\kappa_i-1),m}$ requires at most $O\big(n^3\,\ell^2\,m\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$.

Let us now determine the cost of solving the non-homogeneous linear differential system satisfied by $U_{(i,\kappa_i-1),m}$. For this, we need first the cost of evaluating the matrices $L^{(j)}(\lambda)$ for $j=0,\ldots,\ell$ and $L^{-1}(\lambda)$ (when $\lambda_0+m\notin\sigma(L_0)$) at $\lambda=\lambda_0+m$. This can de done using $O\big((n^2\,\ell^2+n^3\,\ell)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Using the result of Proposition 2.5.3 and bounding p by $n\,\ell$ and m_g by n, solving the system satisfied by $U_{(i,\kappa_i-1),m}$ can be done using at most $O\big(n^4\,\ell^2\,m_a(\lambda_0+m)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$ if $\lambda_0+m\in\sigma(L_0)$, and $O\big(n^3\,\ell^2\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$, otherwise. Since m varies from 1 to ν and i from 1 to $m_g(\lambda_0)$, Step 2.2 can then be performed using at most $O\big(\sum_{m=1}^{\nu}\big(n^4\,\ell^2\,m_a(\lambda_0+m)+n^3\,\ell^2\,m\big)\,m_g(\lambda_0)\,d_{\lambda_0}\big)=O\big(\big(n^5\,\ell^3+n^3\,\ell^2\,\nu^2\big)\,m_g(\lambda_0)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$[4]. Thus, one obtains $m_a(\lambda_0)$ linearly independent regular solutions of exponent λ_0 computed up to order ν in $O\big(\big(n^5\,\ell^3+n^3\,\ell^2\,\nu^2\big)\,m_g(\lambda_0)\,d_{\lambda_0}+n^4\,\ell^3\,m_a(\lambda_0)\,d_{\lambda_0}\big)$ operations in $\mathbb{K}$. Since $\sum_{\lambda_0\in\sigma(L_0)} m_g(\lambda_0)\,d_{\lambda_0}\le\sum_{\lambda_0\in\sigma(L_0)} m_a(\lambda_0)\,d_{\lambda_0}=n\,\ell$, we obtain $O\big(n^4\,\ell^3\,\nu^2+n^6\,\ell^4\big)$ operations in $\mathbb{K}$. □

[3] We omit λ_0 from the notation of the partial multiplicities κ_i for sake of brevity.

[4] We use the fact that $\sum_{m=1}^{\nu} m_a(\lambda_0+m)\le n\,\ell$ with $m_a(\lambda_0+m)=0$ if $\lambda_0+m\notin\sigma(L_0)$.

2.6.3 Second version: by packet

We will give here a slightly modified version of Algorithm **BCE_V1** in which we avoid any computation of partial multiplicities and Jordan chains. We proceed as follows. We first partition the spectrum of $L_0(\lambda)$ into pairwise disjoint subsets $\sigma_1, \ldots, \sigma_r$ such that two eigenvalues belonging to two different subsets σ_i and σ_j do not differ by integers. Then, for each set σ_i for $i \in \{1, \ldots, r\}$, we compute the *general regular solution associated with* σ_i, *i.e.*, the solution

$$y_i(x) = \sum_{\lambda_0 \in \sigma_i} \sum_{j=1}^{m_a(\lambda_0)} c_{\lambda_0,j}\, y_{\lambda_0,j}(x),$$

where the $c_{\lambda_0,j}$'s are arbitrary constants and $y_{\lambda_0,1}(x), \ldots, y_{\lambda_0,m_a(\lambda_0)}(x)$ are $m_a(\lambda_0)$ linearly independent regular formal solutions of (2.1) of exponent λ_0. Choosing $\lambda_i \in \sigma_i$ such that $\Re(\lambda_i) = \min_{\lambda_0 \in \sigma_i} \Re(\lambda_0)$, where $\Re(\lambda_0)$ denotes the real part of the complex number λ_0, the general regular solution associated with σ_i can then be written as

$$y_i(x) = \sum_{m \geq 0} U_{i,m}\, x^{\lambda_i + m}, \tag{2.30}$$

where the $U_{i,m}$'s are n-dimensional vector polynomials in $\log(x)$ depending on arbitrary constants and $U_{i,0} \neq 0$. As we have already seen, λ_i and $U_{i,0}$ must satisfy $L_0(\vartheta)\big(x^{\lambda_i} U_{i,0}\big) = 0$. Hence, to compute $y_i(x)$, we choose $U_{i,0}$ of the form

$$U_{i,0} = \sum_{l=1}^{m_g(\lambda_i)} \sum_{j=0}^{\kappa_l(\lambda_i)-1} C_{i,l,j} \left(\sum_{k=0}^{j} v_{l,j-k}\, \frac{\log^k(x)}{k!} \right), \tag{2.31}$$

where the $C_{i,l,j}$'s are arbitrary constants, the $\kappa_l(\lambda_i)$'s for $l = 1, \ldots, m_g(\lambda_i)$ denote the partial multiplicities at λ_i and $v_{l,0}, \ldots, v_{l,\kappa_l(\lambda_i)-1}$ for $l = 1, \ldots, m_g(\lambda_i)$ form a canonical set of Jordan chains for $L_0(\lambda)$ associated with λ_i (see Proposition 2.4.1). Then, for $m \geq 1$, we choose $U_{i,m}$ as the general polynomial solution in $\log(x)$ given by Corollary 2.5.1 of the non-homogeneous system with constant coefficients

$$L_0(\vartheta + \lambda_i + m)\,(U_{i,m}) = -\sum_{k=0}^{m-1} L_{m-k}(\vartheta + \lambda_i + k)\,(U_{i,k})\,. \tag{2.32}$$

Consider first System (2.32) with $m = 1$. According to Corollary 2.5.1, two cases have to be distinguished: if $\lambda_i + 1$ is not an eigenvalue of $L_0(\lambda)$, then the system has a unique polynomial solution in $\log(x)$: hence, in this case, $U_{i,1}$ only depends on the arbitrary constants appearing in the right-hand side of (2.32), *i.e.*, on the $C_{i,l,j}$'s appearing in the expression of $U_{i,0}$ (see Equation (2.31)). Otherwise, *i.e.*, if $\lambda_i + 1$ is an eigenvalue of $L_0(\lambda)$, then $U_{i,1}$ depends on the $C_{i,l,j}$'s and on $m_a(\lambda_i + 1)$ new arbitrary constants appearing in the general polynomial solution in $t = \log(x)$ of $L_0(\vartheta + \lambda_i + 1)\,(y(x)) = 0$. Hence, continuing this process for $m = 2, 3, \ldots$, if we denote by $\delta_i \in \mathbb{N}^*$ the maximal difference between two elements of σ_i, then U_{i,δ_i} depends on the constants appearing in the right-hand side of (2.32) with $m = \delta_i$ and on $m_a(\lambda_i + \delta_i)$ new arbitrary constants. For $m > \delta_i$, no new arbitrary constants are introduced. Finally, the general regular solution $y_i(x) = \sum_{m \geq 0} U_{i,m}\, x^{\lambda_i + m}$ computed following this way contains $\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)$ arbitrary constants and hence it is the general regular solution associated with the subset σ_i.

Let us now explain how we proceed to compute the vector polynomials $U_{i,m}$ for $m \geq 0$. The vector polynomial $U_{i,0}$ can of course be obtained from a canonical set of Jordan chains for $L_0(\lambda)$ at the eigenvalue λ_i but here we propose to proceed in the same way as we did in

Section 2.5 for computing the general polynomial solution in $t = \log(x)$ for non-homogeneous linear differential systems. From (2.31), we remark that the degree in $t = \log(x)$ of $U_{i,0}$ is equal to $\max\{\kappa_l(\lambda_i), l = 1, \ldots, m_g(\lambda_i)\} - 1$ which can be bounded by $m_a(\lambda_i) - 1$. Hence, let $\alpha_i = m_a(\lambda_i) - 1$ and write $U_{i,0} = \sum_{j=0}^{\alpha_i} U_{(i,0),j} \frac{t^j}{j!}$ where the $U_{(i,0),j}$ are n-dimensional vectors to be determined. Substituting $y(x) = x^{\lambda_i} U_{i,0}$ into $L_0(\vartheta)(y(x)) = 0$ and equating the coefficients of the powers of t to 0 yield the following linear system

$$\begin{pmatrix} L_0(\lambda_i) & & & & \\ \frac{1}{1!}L_0'(\lambda_i) & L_0(\lambda_i) & & (0) & \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ \frac{1}{\alpha_i!}L_0^{(\alpha_i)}(\lambda_i) & \cdots & \cdots & \cdots & L_0(\lambda_i) \end{pmatrix} \begin{pmatrix} U_{(i,0),\alpha_i} \\ U_{(i,0),\alpha_i-1} \\ \vdots \\ \vdots \\ U_{(i,0),0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}.$$

Consequently, computing $U_{i,0}$ is reduced to solving recursively the $\alpha_i + 1 = m_a(\lambda_i)$ linear systems of size n given by

$$\begin{cases} L_0(\lambda_i)\, U_{(i,0),\alpha_i} = 0, \\ L_0(\lambda_i)\, U_{(i,0),j} = -\sum_{k=1}^{\alpha_i - j} \frac{L_0^{(k)}(\lambda_i)}{k!} U_{(i,0),(j+k)}, \quad \text{for} \quad j = \alpha_i - 1, \ldots, 0. \end{cases} \tag{2.33}$$

To solve the above system, we proceed exactly as we did in Section 2.5, *i.e.*, using an LU decomposition of $L_0(\lambda_i)$. Thus, from Proposition 2.5.3, we obtain the complexity of computing $U_{i,0}$ by solving System (2.33).

Lemma 2.6.1. *With the previous notation, computing $U_{i,0}$ by solving System (2.33) can be done using at most $O\big(\big(m_a(\lambda_i)^2 \min\{m_a(\lambda_i), \ell\} + m_a(\lambda_i)^3 + n\big)\, n^2\, d_{\lambda_i}\big)$ operations in $\mathbb{K}$, where d_{λ_i} denotes the degree of the extension $\mathbb{K}(\lambda_i)$ over $\mathbb{K}$.*

Concerning $U_{i,m}$ for $m \geq 1$, we use Algorithm **GenPolSol_NHS** with a slight modification: if $\lambda_i + m$ is an eigenvalue of $L_0(\lambda)$, or, equivalently, 0 is an eigenvalue of $L_0(\lambda + \lambda_i + m)$, we give the algebraic multiplicity of $\lambda_i + m$ as input instead of its partial multiplicities. In this way, the degree bound p becomes

$$\begin{cases} p = d & \text{if} \quad \lambda_i + m \notin \sigma(L_0(\lambda)), \\ p = d + m_a(\lambda_i + m) & \text{if} \quad \lambda_i + m \in \sigma(L_0(\lambda)). \end{cases}$$

Before providing an algorithm derived from the previous discussion and studying its complexity, let us clarify the order of truncation of the power series involved, used in our algorithm.

Remark 2.6.3. Let $\nu \in \mathbb{N}$. If the power series involved in the general regular solution (2.30) associated with σ_i of System (2.1) are truncated at order ν, then we have

$$x^{-\lambda_i}\, \mathcal{L}(x, \vartheta) \left(\sum_{m=0}^{\nu} U_{i,m}\, x^{\lambda_i + m} \right) = 0 \mod x^{\nu+1}.$$

Definition 2.6.1. *Consider a linear differential system of the form (2.1) and an integer $\nu \in \mathbb{N}$. Let $\sigma_1, \ldots, \sigma_r$ be the partition of the spectrum of $L_0(\lambda)$ as defined above. For $i = 1, \ldots, r$, let y_i be the general regular solution associated with σ_i of System (2.1), computed up to order ν. Then, we will say that $y = \sum_{i=1}^{r} y_i$ is the* general regular solution up to order ν of System (2.1).

From the method sketched above, we deduce the following algorithm.

ALGORITHM **BCE_V2**

INPUT: An integer $\nu \in \mathbb{N}$ and the coefficient matrices $A_i(x)$ of System (2.1) truncated at order ν (see Remark 2.6.1).
OUTPUT: The general regular solution up to order ν of System (2.1).

INITIALIZATION: Define $L_0(\lambda) = \sum_{i=0}^{\ell} A_i(0)\,\lambda^i$;

1. Compute $\sigma(L_0)$ and partition it into subsets $\sigma_1, \ldots, \sigma_r$ defined as above;
   // see Remark 2.6.2
2. **For** i from 1 to r **do**
   2.1. Let $\lambda_i \in \sigma_i$ be such that $\Re(\lambda_i) = \min_{\lambda_0 \in \sigma_i} \Re(\lambda_0)$;
   2.2. Compute $U_{i,0}$ by solving the system given by (2.33);
   2.3. **For** m from 1 to ν **do**
   Compute $U_{i,m}$ as the general polynomial solution of System (2.32);
   end do;
   2.4. Let $y_i = \sum_{m=0}^{\nu} U_{i,m}\, x^{\lambda_i+m}$;
   end do;
3. **Return** $y = \sum_{i=1}^{r} y_i$;

Proposition 2.6.2. *The algorithm* **BCE_V2** *returns the general regular solution up to order ν of system (2.1) after at most $O\big(n^4\,\ell^3\,\nu^2 + n^6\,\ell^4\big)$ operations in $\mathbb{K}$.*

Proof. We first compute the matrix inverse $L_0^{-1}(\lambda)$ and the first ℓ derivatives of the matrices $L_i(\lambda)$ for $i = 0, \ldots, \nu$. This costs at most $O^{\sim}\big(n^3\,\ell + n^2\,\ell^2\,\nu\big)$ operations in $\mathbb{K}$ (see the proof of Proposition 2.6.1). Let now λ_i and σ_i be as defined in the algorithm. According to Lemma 2.6.1, Step 2.2 can be performed using at most $O\big(\big(m_a(\lambda_i)^2\,\min\{m_a(\lambda_i), \ell\} + m_a(\lambda_i)^3 + n\big)\,n^2\,d_{\lambda_i}\big) = O\big(n^5\,\ell^3\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Let now determine the cost of Step 2.3. We first consider the cost of computing the right-hand side of System (2.32). Since the $U_{i,j}$ for $i = 0, \ldots, m-1$ depend on arbitrary constants whose number do not exceed $\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)$, following the computation done in the proof of Proposition 2.6.1 we find that computing the right-hand side of System (2.32) can be done using at most $O\big(\big(\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\big)\,n^3\,\ell^2\,m\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Now using the result of Proposition 2.5.3 and bounding p by $n\,\ell$ and m_g by n, the complexity of solving System (2.32) is bounded by $O\big(\big(\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\big)\,n^4\,\ell^2\,m_a(\lambda_i + m)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$[5] if $\lambda_i + m \in \sigma(L_0)$, and $O\big(\big(\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\big)\,n^3\,\ell^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$, otherwise. Thus, Step 2.3 costs $O\big(\big(n^3\,\ell^2\,\nu^2 + n^5\,\ell^3\big)\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$[6]. Since $\sum_{i=1}^{r}\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\,d_{\lambda_i} = n\,\ell$, Algorithm **BCE_V2** returns the general regular solution up to order ν of system (2.1) using $O\big(n^4\,\ell^3\,\nu^2 + n^6\,\ell^4\big)$ operations in $\mathbb{K}$. □

Algorithm **BCE_V2** above has been implemented[7] in MAPLE using the LINEARALGEBRA package.

2.7 Generalization of Frobenius' method

A classical method for computing a basis of the regular formal solution space of scalar linear differential equations of the form (2.1) (with $n = 1$) is Frobenius' method (see [37, Chap. 4,

[5] Since the right-hand side of the system depends on at most $\sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)$ arbitrary constants.
[6] We use the fact that $\sum_{m=1}^{\nu} m_a(\lambda_0 + m) \le n\,\ell$ with $m_a(\lambda_0 + m) = 0$ if $\lambda_0 + m \notin \sigma(L_0)$.
[7] The implementation is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`

Section 8], [44], [59, Chap. 16] or [94, Chap. 2]). This method has been generalized to first-order linear differential systems of the first kind in [92, Chap. 3] (see also [54, Chap. 4]). As we have seen in Section 2.3, there exist some works [64, 81, 80] adapting Frobenius' method to higher-order linear differential systems, but these works do not treat the general case. Indeed, generalizing Frobenius' method to the matrix case is not a simple task. Nevertheless, based on the theory of matrix polynomials, we manage to propose a generalization of the Frobenius method to handle arbitrary systems of the form (2.1). Our approach follows the broad outlines of the Frobenius method in the scalar case.

We consider an n-dimensional vector of logarithm-free formal power series of the form

$$y(x, \lambda, g_0) = \sum_{i=0}^{\infty} g_i(\lambda)\, x^{\lambda+i}, \tag{2.34}$$

where $\forall\, i \geq 0$, $g_i \in \mathbb{K}(\lambda)^n$ and $g_0 \neq 0$, and we seek to determine the g_i's as vectors of rational functions in λ in such a way that $y(x, \lambda, g_0)$ is a nontrivial solution of the non-homogeneous differential system

$$\mathcal{L}(x, \vartheta)(y(x)) = L_0(\lambda)\, g_0(\lambda)\, x^{\lambda}. \tag{2.35}$$

Since

$$\forall\, j \geq 0, \quad \vartheta^j \left(\sum_{i=0}^{\infty} g_i(\lambda)\, x^{\lambda+i} \right) = \sum_{i=0}^{\infty} (\lambda+i)^j g_i(\lambda)\, x^{\lambda+i},$$

substituting (2.34) into $\mathcal{L}(x, \vartheta)$ yields

$$\mathcal{L}(x, \vartheta)(y(x, \lambda, g_0)) = \sum_{i=0}^{\infty} \left(\sum_{j=0}^{i} L_j(\lambda+i-j)\, g_{i-j}(\lambda) \right) x^{\lambda+i},$$

where the L_j's are given by (2.23). Identifying the coefficients of the powers of x, we find that $y(x, \lambda, g_0)$ is a solution of System (2.35) if and only if

$$\forall\, i \geq 1, \quad L_0(\lambda+i)\, g_i(\lambda) = -\sum_{j=1}^{i} L_j(\lambda+i-j)\, g_{i-j}(\lambda).$$

As $L_0(\lambda)$ is a regular matrix polynomial, *i.e.*, $\det(L_0(\lambda)) \neq 0$, the above systems can be solved recursively for $g_1(\lambda), g_2(\lambda), \ldots$ as vectors of rational functions. Indeed, taking $g_0(\lambda)$ as an arbitrary n-dimensional vector of rational functions in λ and supposing that $g_1(\lambda), \ldots, g_{i-1}(\lambda)$ have already been determined, $g_i(\lambda)$ is given by

$$g_i(\lambda) = -L_0^{-1}(\lambda+i) \sum_{j=1}^{i} L_j(\lambda+i-j)\, g_{i-j}(\lambda). \tag{2.36}$$

Hence, for $i \geq 1$, the vector g_i of rational functions is well-defined at any element λ_0 of $\overline{\mathbb{K}}$ where g_0 is well-defined and $\det(L_0(\lambda_0+i)) \neq 0$, *i.e.*, $\lambda_0 + i$ is not an eigenvalue of $L_0(\lambda)$. In other terms, if one chooses λ_0 as an element of $\overline{\mathbb{K}}$ such that $\forall\, i \geq 1$, $\lambda_0 + i$ is not an eigenvalue of $L_0(\lambda)$ and chooses g_0 as an element of $\mathbb{K}(\lambda)^n$ defined at λ_0, then $y(x, \lambda, g_0)$ is well-defined at λ_0. For $y(x, \lambda_0, g_0)$ to be a nontrivial solution of System (2.1), two other conditions are required:

1. $g_0(\lambda_0)$ must be a nonzero vector, and
2. the right-hand side of (2.35) must vanish for $\lambda = \lambda_0$, *i.e.*, $L_0(\lambda_0)\, g_0(\lambda_0) = 0$.

These conditions are ensured if one takes λ_0 as an eigenvalue of $L_0(\lambda)$, and $g_0(\lambda_0)$ as an eigenvector of $L_0(\lambda)$ associated with λ_0. To summarize, by taking $\lambda_0 \in \sigma(L_0)$ such that $\forall i \geq 1$, $\lambda_0 + i \notin \sigma(L_0)$ and taking $g_0 \in \mathbb{K}(\lambda)^n$ such that $g_0(\lambda_0)$ is an eigenvector of $L_0(\lambda)$ associated with λ_0 (*e.g.*, one can take $g_0(\lambda)$ as a root polynomial of $L_0(\lambda)$ associated with λ_0, or $g_0(\lambda) = v$, where v is an eigenvector of $L_0(\lambda)$ associated with λ_0), one has a logarithm-free regular solution of (2.1) given by $y(x, \lambda_0, g_0)$.

Theorem 2.7.1. *If λ_0 is an eigenvalue of the matrix polynomial $L_0(\lambda) = \sum_{i=0}^{\ell} A_i(0)\,\lambda^i$ such that $\lambda_0 + i \notin \sigma(L_0)$ for every positive integer i, then one can construct $m_g(\lambda_0)$ linearly independent logarithm-free regular solutions of System (2.1) of exponent λ_0 of the form $y(x, \lambda_0, g_0)$, where $y(x, \lambda, g_0)$ is given by (2.34) with $g_0(\lambda_0)$ an eigenvector of $L_0(\lambda)$ associated with λ_0.*

The two following corollaries are direct consequences of the theorem above.

Corollary 2.7.1. *Consider a system of the form (2.1). If the matrix polynomial defined by $L_0(\lambda) = \sum_{i=0}^{\ell} A_i(0)\,\lambda^i$ has $n\,\ell$ distinct eigenvalues which do not differ by integers, then System (2.1) admits $n\,\ell$ linearly independent logarithm-free regular solutions of the form $y(x, \lambda_0, g_0)$, where $y(x, \lambda, g_0)$ is given by (2.34), $\lambda_0 \in \sigma(L_0)$ and $g_0(\lambda_0)$ is an eigenvector of $L_0(\lambda)$ associated with λ_0.*

Corollary 2.7.2. *Consider a system of the form (2.1). If the distinct eigenvalues of the matrix polynomial $L_0(\lambda) = \sum_{i=0}^{\ell} A_i(0)\,\lambda^i$ do not differ by integers and if for every $\lambda_0 \in \sigma(L_0)$, one has $m_a(\lambda_0) = m_g(\lambda_0)$[8], then System (2.1) admits $n\,\ell$ linearly independent logarithm-free regular solutions of the form $y(x, \lambda_0, g_0)$, where $y(x, \lambda, g_0)$ is given by (2.34), $\lambda_0 \in \sigma(L_0)$ and $g_0(\lambda_0)$ is an eigenvector of $L_0(\lambda)$ associated with λ_0.*

The difficulty arises then when one of the following cases happens:

- **First case:** The matrix polynomial $L_0(\lambda)$ has an eigenvalue λ_0 such that $\forall\, i \geq 1$, $\lambda_0 + i \notin \sigma(L_0)$ and one of the partial multiplicities of $L_0(\lambda)$ at λ_0 is greater than or equal to 2.
- **Second case:** The matrix polynomial $L_0(\lambda)$ has an eigenvalue λ_0 such that $\lambda_0 + i$ with $i \in \mathbb{N}^*$ is also an eigenvalue of $L_0(\lambda)$.

In the sequel, we explain how to compute $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0 in each of the two cases described above.

2.7.1 First case

We suppose here that $\forall\, i \geq 1$, $\lambda_0 + i$ is not an eigenvalue of $L_0(\lambda)$ and that there exists a partial multiplicity κ of $L_0(\lambda)$ at λ_0 such that $\kappa \geq 2$. One can compute κ linearly independent regular solutions of (2.1) of exponent λ_0 by proceeding in the following way. Differentiating formally Equation (2.35) with respect to the parameter λ, we find

$$\forall\, k \geq 1, \quad \frac{\partial^k}{\partial \lambda^k}\mathcal{L}(x, \vartheta)(y(x, \lambda, g_0)) = \sum_{j=0}^{k} \binom{k}{j} \frac{\partial^j}{\partial \lambda^j}\left(L_0(\lambda) g_0(\lambda)\right) \log^{k-j}(x)\, x^{\lambda},$$

where $\binom{k}{j} = \frac{k!}{j!\,(k-j)!}$ is a binomial coefficient. Taking into account the commutativity of the two derivations $\frac{\partial}{\partial \lambda}$ and ϑ, we have

$$\frac{\partial^k}{\partial \lambda^k}\mathcal{L}(x, \vartheta)(y(x, \lambda, g_0)) = \mathcal{L}(x, \vartheta)\left(\frac{\partial^k y}{\partial \lambda^k}(x, \lambda, g_0)\right),$$

[8] This is equivalent to saying that all the partial multiplicities of $L_0(\lambda)$ at λ_0 are equal to 1.

and hence

$$\forall\, k \geq 1, \quad \mathcal{L}(x,\vartheta)\left(\frac{\partial^k y}{\partial \lambda^k}(x,\lambda,g_0)\right) = \sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \frac{\partial^j}{\partial \lambda^j}\left(L_0(\lambda)g_0(\lambda)\right) \log^{k-j}(x)\, x^{\lambda}. \tag{2.37}$$

Since $\kappa \geq 2$, it is always possible to choose $g_0(\lambda)$ as a root polynomial of $L_0(\lambda)$ of maximal order κ associated with eigenvalue λ_0. For this choice of g_0, the right-hand side of (2.37) vanishes for $\lambda = \lambda_0$ and $k = 1, \ldots, \kappa - 1$. Therefore, the vectors

$$\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_0) = \sum_{i=0}^{\infty}\sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \frac{\partial^j g_i}{\partial \lambda^j}(\lambda_0) \log^{k-j}(x)\, x^{\lambda_0+i}, \quad \text{for } k = 0, \ldots, \kappa - 1 \tag{2.38}$$

form κ regular solutions of System (2.1).

Lemma 2.7.1. *The vectors $\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_0)$ given in (2.38) with $g_0(\lambda)$ a root polynomial of $L_0(\lambda)$ of maximal order κ associated with the eigenvalue λ_0 form κ regular solutions of exponent λ_0 linearly independent over $\overline{\mathbb{K}}$.*

Proof. Notice that the right-hand side of the equality in (2.38) contains the nonzero term $g_0(\lambda_0)\log^k(x)\,x^{\lambda_0}$. Hence, every solution $\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_0)$ can be written as $\sum_{i\geq 0} U_{k,i}\, x^{\lambda_0+i}$ with $U_{k,i} \in \overline{\mathbb{K}}[\log(x)]^n$ and $U_{k,0} \neq 0$. Thus, the solutions $\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_0)$ for $k = 0, \ldots, \kappa-1$ are regular solutions of exponent λ_0. Now, the linearly independency follows from the fact that the $U_{k,0}$ for $k = 0, \ldots, \kappa - 1$ are of strictly increasing degrees in $\log(x)$. □

The following proposition shows how to construct $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0 where $\lambda_0 \in \sigma(L_0)$ such that $\forall\, i \in \mathbb{N}^*$, $\lambda_0 + i \notin \sigma(L_0)$.

Proposition 2.7.1. *Let λ_0 be an eigenvalue of $L_0(\lambda)$ such that $\forall\, i \in \mathbb{N}^*$, $\lambda_0 + i \notin \sigma(L_0)$. Let $\kappa_1, \ldots, \kappa_{m_g(\lambda_0)}$ denote the partial multiplicities of $L_0(\lambda)$ at λ_0. For $i = 1, \ldots, m_g(\lambda_0)$, define $g_{i,0}(\lambda)$ as a root polynomial of $L_0(\lambda)$ of maximal order κ_i associated with λ_0 such that the eigenvectors $g_{1,0}(\lambda_0), \ldots, g_{m_g(\lambda_0),0}(\lambda_0)$ are linearly independent [9] over $\overline{\mathbb{K}}$. The vectors*

$$\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_{i,0}) \quad \text{for } i = 1, \ldots, m_g(\lambda_0) \text{ and } k = 0, \ldots, \kappa_i - 1,$$

where $y(x,\lambda,g_{i,0})$ is given by (2.34) with $g_0 = g_{i,0}$, form $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0.

Proof. For a fixed integer $i \in \{1, \ldots, m_g(\lambda_0)\}$, the vectors $\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_{i,0})$ for $k = 0, \ldots, \kappa_i - 1$ are linearly independent over $\overline{\mathbb{K}}$ due to Lemma 2.7.1. For $i_1, i_2 \in \{1, \ldots, m_g(\lambda_0)\}$ such that $i_1 \neq i_2$, the two vectors $\frac{\partial^k y}{\partial \lambda^k}(x,\lambda_0,g_{i_1,0})$ and $\frac{\partial^j y}{\partial \lambda^j}(x,\lambda_0,g_{i_2,0})$ are linearly independent over $\overline{\mathbb{K}}$ since the eigenvectors $g_{i_1,0}(\lambda_0)$ and $g_{i_2,0}(\lambda_0)$ are so. □

2.7.2 Second case

Suppose that there exist r $(r \geq 1)$ positive integers $n_1 < \cdots < n_r$ such that $\lambda_0 + n_i$ is an eigenvalue of $L_0(\lambda)$ and $\lambda_0 + i$ is not for $i \in \mathbb{N}^* \setminus \{n_1, \ldots, n_r\}$. For $i = 1, \ldots, r$, let m_i denote the algebraic multiplicity of $\lambda_0 + n_i$ and let $m = \sum_{i=1}^{r} m_i$. We explain in the sequel how one can construct $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0. We proceed

[9] This means that the Jordan chains derived from $g_{i,0}(\lambda)$ for $i = 1, \ldots, m_g(\lambda_0)$ form a canonical set of Jordan chains for $L_0(\lambda)$ associated with λ_0.

as earlier, *i.e.*, we seek the solutions of (2.1) from those of non-homogeneous systems properly chosen.

Instead of (2.35), we consider the non-homogeneous differential system

$$\mathcal{L}(x,\vartheta)(y(x)) = (\lambda-\lambda_0)^m\, L_0(\lambda)\, g_0(\lambda)\, x^\lambda, \tag{2.39}$$

where $g_0(\lambda)$ is an arbitrary n-dimensional vector of rational functions well-defined at $\lambda=\lambda_0$ and we look for logarithm-free regular formal solutions of (2.39) of the form

$$y(x,\lambda,g_0) = (\lambda-\lambda_0)^m\, g_0(\lambda)\, x^\lambda + \sum_{i=1}^{\infty} g_i(\lambda)\, x^{\lambda+i}, \tag{2.40}$$

where, for $i\geq 1$, $g_i(\lambda)$ is an n-dimensional vector of rational functions to be determined. Plugging (2.40) into (2.39) yields

$$L_0(\lambda+1)\, g_1(\lambda) = -(\lambda-\lambda_0)^m\, L_1(\lambda)\, g_0(\lambda),$$

$$\text{and for } i\geq 2,\quad L_0(\lambda+i)\, g_i(\lambda) = -\sum_{j=1}^{i-1} L_j(\lambda+i-j)\, g_{i-j}(\lambda) - (\lambda-\lambda_0)^m\, L_i(\lambda)\, g_0(\lambda). \tag{2.41}$$

For $i=1,\dots,n_1-1$, we have $\det(L_0(\lambda_0+i))\neq 0$, thus $g_i(\lambda)$ is well-defined at λ_0. Furthermore, notice that $g_i(\lambda)$ can be written as $g_i(\lambda)=(\lambda-\lambda_0)^m\, M_i(\lambda)\, g_0(\lambda)$ with $M_i(\lambda)\in\mathbb{K}(\lambda)^{n\times n}$ well-defined at $\lambda=\lambda_0$, so $g_i(\lambda_0)=0$ for $i=1,\dots,n_1-1$.

For $i=n_1$ in (2.41), the right-hand side of (2.41) can be written as $(\lambda-\lambda_0)^m\, F_{n_1}(\lambda)\, g_0(\lambda)$ with $F_{n_1}(\lambda)\in\mathbb{K}(\lambda)^{n\times n}$ well-defined at $\lambda=\lambda_0$. Therefore, $g_{n_1}(\lambda)$ can be written as

$$\begin{aligned} g_{n_1}(\lambda) &= (\lambda-\lambda_0)^m\, L_0^{-1}(\lambda+n_1)\, F_{n_1}(\lambda)\, g_0(\lambda) \\ &= \frac{(\lambda-\lambda_0)^m}{\det(L_0(\lambda+n_1))}\operatorname{adj}(L_0(\lambda+n_1))\, F_{n_1}(\lambda)\, g_0(\lambda), \end{aligned} \tag{2.42}$$

where $\operatorname{adj}(L_0(\lambda+n_1))$ denotes the adjoint (the transpose of the cofactor matrix) of $L_0(\lambda+n_1)$. Since λ_0+n_1 is a root of multiplicity m_1 for $\det(L_0(\lambda))$, λ_0 is a root of multiplicity m_1 for $\det(L_0(\lambda+n_1))$, *i.e.*, $\det(L_0(\lambda+n_1)) = (\lambda-\lambda_0)^{m_1}\, p(\lambda)$ with $p(\lambda)\in\mathbb{K}[\lambda]$ such that $p(\lambda_0)\neq 0$. Therefore, after simplification in (2.42), the expression of $g_{n_1}(\lambda)$ becomes

$$g_{n_1}(\lambda) = \frac{(\lambda-\lambda_0)^{m-m_1}}{p(\lambda)}\operatorname{adj}(L_0(\lambda+n_1))\, F_{n_1}(\lambda)\, g_0(\lambda)$$

and hence it is well-defined at $\lambda=\lambda_0$. Continuing with the same process, we find that

$$\begin{cases} g_i(\lambda) = (\lambda-\lambda_0)^{m-m_1}\, M_i(\lambda)\, g_0(\lambda), & \text{for } i=n_1,\dots,n_2-1,\\ g_i(\lambda) = (\lambda-\lambda_0)^{m-m_1-m_2}\, M_i(\lambda)\, g_0(\lambda), & \text{for } i=n_2,\dots,n_3-1,\\ \qquad\vdots & \\ g_i(\lambda) = (\lambda-\lambda_0)^{m_r}\, M_i(\lambda)\, g_0(\lambda), & \text{for } i=n_{r-1},\dots,n_r-1,\\ g_i(\lambda) = M_i(\lambda)\, g_0(\lambda), & \text{for } i\geq n_r, \end{cases}$$

where $M_i(\lambda)\in\mathbb{K}(\lambda)^{n\times n}$ is well-defined at $\lambda=\lambda_0$. It follows that for $i=1,\dots,n_r-1$, $g_i(\lambda_0)=0$ and substituting λ by λ_0 in (2.40) gives

$$y(x,\lambda_0,g_0) = \sum_{i=n_r}^{\infty} g_i(\lambda_0)\, x^{\lambda_0+i} = \sum_{i=0}^{\infty} g_{i+n_r}(\lambda_0)\, x^{\lambda_0+n_r+i}.$$

From (2.41), the vector $g_{n_r}(\lambda_0)$ satisfies $L_0(\lambda_0+n_r)\,g_{n_r}(\lambda_0)=0$ and hence $g_{n_r}(\lambda_0)$ is either a zero vector or an eigenvector associated with λ_0+n_r. In both cases, $y(x,\lambda_0,g_0)$ is linearly dependent to the regular solutions of (2.1) of exponent λ_0+n_r which can be computed as explained in Proposition 2.7.1. Nevertheless, since we search for regular solutions that are linearly independent from those of exponent $\lambda_0+n_1,\dots,\lambda_0+n_r$, we differentiate formally Equation (2.39) m times with respect to λ. We find

$$\mathcal{L}(x,\vartheta)\left(\frac{\partial^m y}{\partial\lambda^m}(x,\lambda,g_0)\right)=\sum_{j=0}^{m}\binom{m}{j}\frac{\partial^j}{\partial\lambda^j}\left((\lambda-\lambda_0)^m\right)\frac{\partial^{m-j}}{\partial\lambda^{m-j}}\left(L_0(\lambda)\,g_0(\lambda)\,x^\lambda\right). \tag{2.43}$$

Therefore, $\frac{\partial^m y}{\partial\lambda^m}(x,\lambda_0,g_0)$ is a regular solution of the homogeneous system $\mathcal{L}(x,\vartheta)(y(x))=0$ if and only if the right-hand side of (2.43) vanishes for $\lambda=\lambda_0$, *i.e.*, if and only if $L_0(\lambda_0)\,g_0(\lambda_0)=0$ since $\frac{\partial^j}{\partial\lambda^j}\left((\lambda-\lambda_0)^m\right)|_{\lambda=\lambda_0}=0$ for $j=0,\dots,m-1$. So one has to choose $g_0(\lambda)$ such that $g_0(\lambda_0)$ is an eigenvector of $L_0(\lambda)$ associated with λ_0. Moreover, for this choice of $g_0(\lambda)$, $\frac{\partial^m y}{\partial\lambda^m}(x,\lambda_0,g_0)$ is a regular solution of exponent λ_0 since it has a nonzero term of the form $m!\,g_0(\lambda_0)\,x^{\lambda_0}$.

Hence, one can construct $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0 as following.

Proposition 2.7.2. *Let λ_0 be an eigenvalue of $L_0(\lambda)$ and $\kappa_1,\dots,\kappa_{m_g(\lambda_0)}$ denote the partial multiplicities of $L_0(\lambda)$ at λ_0. For $i=1,\dots,m_g(\lambda_0)$, define $g_{i,0}(\lambda)$ as a root polynomial of $L_0(\lambda)$ of maximal order κ_i associated with λ_0 such that the vectors $g_{1,0}(\lambda_0),\dots,g_{m_g(\lambda_0),0}(\lambda_0)$ are linearly independent over $\overline{\mathbb{K}}$. Suppose that there exist $n_1,\dots,n_r\in\mathbb{N}^*$ such that λ_0+n_i is an eigenvalue of $L_0(\lambda)$ and $\forall\, i\in\mathbb{N}^*\setminus\{n_1,\dots,n_r\}$, λ_0+i is not. Set $m=\sum_{i=1}^r m_a(\lambda_0+n_i)$. The vectors*

$$\frac{\partial^{m+k}y}{\partial\lambda^{m+k}}(x,\lambda_0,g_{i,0})\quad \text{for } i=1,\dots,m_g(\lambda_0) \text{ and } k=0,\dots,\kappa_i-1,$$

form $m_a(\lambda_0)$ linearly independent regular solutions of (2.1) of exponent λ_0. Here, $y(x,\lambda,g_{i,0})$ is given by (2.40) with $g_0=g_{i,0}$,

Proof. For each $i\in\{1,\dots,m_g(\lambda_0)\}$, the $(m+k)$th derivative $(0\le k\le\kappa_i-1)$ of the right-hand side of (2.39) for $g_0=g_{i,0}$ can be written as (I)+(II), where

$$\text{(I)}=\sum_{j=0}^{m}\binom{m+k}{j}\frac{\partial^j}{\partial\lambda^j}\left((\lambda-\lambda_0)^m\right)\frac{\partial^{m+k-j}}{\partial\lambda^{m+k-j}}\left(L_0(\lambda)\,g_{i,0}(\lambda)\,x^\lambda\right),$$

and

$$\begin{aligned}\text{(II)}&=\sum_{j=m+1}^{m+k}\binom{m+k}{j}\frac{\partial^j}{\partial\lambda^j}\left((\lambda-\lambda_0)^m\right)\frac{\partial^{m+k-j}}{\partial\lambda^{m+k-j}}\left(L_0(\lambda)\,g_{i,0}(\lambda)\,x^\lambda\right)\\&=\sum_{j=1}^{k}\binom{m+k}{m+j}\frac{\partial^{m+j}}{\partial\lambda^{m+j}}\left((\lambda-\lambda_0)^m\right)\frac{\partial^{k-j}}{\partial\lambda^{k-j}}\left(L_0(\lambda)\,g_{i,0}(\lambda)\,x^\lambda\right).\end{aligned}$$

(I) evaluated at $\lambda=\lambda_0$ is equal to

$$m!\,\frac{\partial^k}{\partial\lambda^k}\left(L_0(\lambda)\,g_{i,0}(\lambda)\,x^\lambda\right)_{\lambda=\lambda_0}$$

which is zero by definition of $g_{i,0}(\lambda)$. Similarly, (II) vanishes for $\lambda=\lambda_0$ since $\frac{\partial^{m+j}}{\partial\lambda^{m+j}}\left((\lambda-\lambda_0)^m\right)=0$ for $j\ge1$. Thus, the $(m+k)$th derivative of the right-hand side of (2.39) for $g_0=g_{i,0}$ vanishes

for $\lambda = \lambda_0$. Therefore, $\frac{\partial^{m+k} y}{\partial \lambda^{m+k}}(x, \lambda_0, g_{i,0})$ for $i = 1, \ldots, m_g(\lambda_0)$ and $k = 0, \ldots, \kappa_i - 1$ are regular solutions of (2.1). Furthermore, it is easy to check that each solution $\frac{\partial^{m+k} y}{\partial \lambda^{m+k}}(x, \lambda_0, g_{i,0})$ can be written as $\sum_{j \geq 0} U_{i,k,j}\, x^{\lambda_0 + j}$ with $U_{i,k,j} \in \overline{\mathbb{K}}[\log(x)]^n$ and

$$U_{i,k,0} = \left(\begin{array}{c} m+k \\ m \end{array} \right) m! \sum_{s=0}^{k} \left(\begin{array}{c} k \\ s \end{array} \right) \frac{\partial^s g_{i,0}}{\partial \lambda^s}(\lambda_0)\, \log^{k-s}(x).$$

Thus, $U_{i,k,0}$ is of degree k in $\log(x)$ since $g_{i,0}(\lambda_0) \neq 0$. Consequently, the vectors $\frac{\partial^{m+k} y}{\partial \lambda^{m+k}}(x, \lambda_0, g_{i,0})$ for $i = 1, \ldots, m_g(\lambda_0)$ and $k = 0, \ldots, \kappa_i - 1$ are regular solutions of (2.1) of exponent λ_0 and they are linearly independent over $\overline{\mathbb{K}}$ since the $U_{i,k,0}$'s are so (recall that $g_{i_1,0}(\lambda_0)$ and $g_{i_2,0}(\lambda_0)$ are linearly independent for $i_1 \neq i_2$). □

2.7.3 Summary and example

We sum up our generalization of the Frobenius method for computing regular formal solutions of System (2.1) as follows. First, compute $\sigma(L_0)$ and partition it into r pairwise disjoint subsets $\sigma_1, \ldots, \sigma_r$ such that two eigenvalues belonging to two different subsets σ_i and σ_j do not differ by integers. For $i = 1, \ldots, r$, let $\lambda_i \in \sigma_i$ such that $\Re(\lambda_i) = \min_{\lambda_0 \in \sigma_i} \Re(\lambda_0)$. If $m_a(\lambda_i) = m_g(\lambda_i)$, then one can compute $m_a(\lambda_i)$ linearly independent logarithm-free regular solutions of System (2.1) of exponent λ_i as in Theorem 2.7.1. Otherwise, one can have $m_a(\lambda_i)$ linearly independent regular solutions of exponent λ_i by proceeding as in Proposition 2.7.1. For $\lambda_0 \in \sigma_i$ different from λ_i, one can compute $m_a(\lambda_0)$ linearly independent regular solutions of exponent λ_0 by proceeding as in Proposition 2.7.2. In this way, one can construct a basis of the regular formal solution space of systems of the first kind of the form (2.1), composed of $\deg(\det(L_0(\lambda))) = n\ell$ linearly independent regular solutions.

We have implemented[10] the generalization of Frobenius' method described above in MAPLE using the LINEARALGEBRA package. Our implementation computes a basis of the regular formal solution space of a system of the form (2.1), where the series involved are computed up to a fixed order ν given as input.

Example 2.7.1. *We consider an application taken from [71]. The incipient buoyant thermal convection in vertical cylindrical geometries (see Equations (2.17)-(2.19) of [71]) gives rise to the system composed of* 3 *linear differential equations with variable* r *and unknowns* ψ, $\overline{p}$ *and* $\overline{T}$

$$-r^2\, \vartheta(\overline{p}) + m\, \vartheta^2(\psi) - 2\, m\, \vartheta(\psi) - m^3\, r^2\, \psi = 0, \tag{2.44}$$

$$m\, r^4\, \overline{p} - \vartheta^3(\psi) + 4\, \vartheta^2(\psi) + \left(m^2\, r^2 - 4\right) \vartheta(\psi) + \mathrm{Ra}\, r^4\, \overline{T} = 0, \tag{2.45}$$

$$\vartheta^2(\overline{T}) - m^2\, r^2\, \overline{T} - \vartheta(\psi) = 0, \tag{2.46}$$

where $\vartheta = r\, \frac{d}{dr}$ *and* m *and* Ra *are two nonzero constants. In order to compute regular solutions of this system, the authors of [71] reduce it to a scalar linear differential equation in* ψ *of order* 6 *and then apply the classical Frobenius method. Here, we propose to apply our generalization of Frobenius' method to the system directly. But before, we will transform the system into another one which will be of the first kind. Multiplying Equation (2.45) on the left by* $r^{-2}\, m$ *then adding to the obtained equation, Equation (2.44) multiplied on the left by the scalar operator* $r^{-2}\, (\vartheta - 2)$ *yield an equivalent system (see Chapter 3, Subsection 3.6.1) of second-order which can be written*

[10] The program is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`

as

$$\mathcal{L}(r,\vartheta)(y)=\begin{pmatrix} m & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{pmatrix}\vartheta^2(y)+\begin{pmatrix} -2\,m & -r^2 & 0\\ 0 & 0 & 0\\ -1 & 0 & 0\end{pmatrix}\vartheta(y)+\begin{pmatrix} -m^3\,r^2 & 0 & 0\\ 0 & m^2\,r^2 & \mathrm{Ra}\,m\,r^2\\ 0 & 0 & -m^2\,r^2\end{pmatrix}y=0, \tag{2.47}$$

where $y=\begin{pmatrix}\psi & \overline{p} & \overline{T}\end{pmatrix}^T$ *is the vector of unknowns. System (2.47) is of the first kind since its leading coefficient is an invertible constant matrix. The matrix polynomial* $L_0(\lambda)$ *associated with System (2.47) is given by*

$$L_0(\lambda)=\begin{pmatrix} m\,\lambda^2-2\,m\,\lambda & 0 & 0\\ 0 & -\lambda^2 & 0\\ -\lambda & 0 & \lambda^2\end{pmatrix}$$

and has 2 eigenvalues: 0 *of algebraic multiplicity* $m_a(0)=5$ *and* 2 *of algebraic multiplicity* $m_a(2)=1$. *Thus, a basis of the regular formal solution space of (2.47) is composed of:*

- *one logarithm-free regular solution of exponent* 2 *of the form* $y(x,\lambda,g_0)$ *given by (2.34) with* $\lambda=2$ *and* $g_0(2)$ *an eigenvector of* $L_0(\lambda)$ *associated with* 2, *and*
- 5 *regular solutions of exponent* 0 *which can be obtained following the indications in Proposition 2.7.2. Here, we note that the eigenvalue* 0 *has three partial multiplicities* $\kappa_1=1$, $\kappa_2=2$ *and* $\kappa_3=2$.

In the sequel, we will use our MAPLE *implementation of the generalization of the Frobenius method to compute a basis of the regular solution space of System (2.47). For this, we need to define the coefficient matrices of the system*

$$A2:=\begin{bmatrix} m & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}\quad A1:=\begin{bmatrix} -2\,m & -r^2 & 0\\ 0 & 0 & 0\\ -1 & 0 & 0\end{bmatrix}\quad A0:=\begin{bmatrix} -m^3\,r^2 & 0 & 0\\ 0 & m^2\,r^2 & \mathrm{Ra}\,m\,r^2\\ 0 & 0 & -m^2\,r^2\end{bmatrix}.$$

Taking $\nu=2$, *we obtain a basis of the regular formal solution space of System (2.47) composed of vectors:*

```
> Frobenius([A2,A1,A0],r,{Ra,m},2);
```

$$\begin{bmatrix} 2\,r^2\left(8+m^2\,r^2\right)\\ \frac{1}{2}\,r^4\,\mathrm{Ra}\,m\\ r^2\left(8+m^2\,r^2\right)\end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2}\,\frac{4\,m+2\,r^2\,\log(r)\,m^3+10\,r^2-m^3\,r^2}{m}\\ 10+\frac{5}{2}\,m^2\,r^2+\frac{5}{4}\,\mathrm{Ra}\,m\,r^2\\ \frac{1}{4}\,\frac{20\,m+2\,r^2\,\log(r)\,m^3+10\,r^2+3\,m^3\,r^2}{m}\end{bmatrix}$$

$$\begin{bmatrix} \frac{r^2}{m}\\ 2+\frac{1}{2}\,m^2\,r^2+\mathrm{Ra}\,m\,r^2\\ \frac{1}{2}\,\frac{8\,m+r^2+2\,m^3\,r^2}{m}\end{bmatrix}$$

$$\begin{bmatrix} \frac{r^2(2\,\log(r)-1)}{m} \\ 4\,\log(r)+r^2\,\log(r)\,m^2+2\,r^2\,\log(r)\,m\,\mathrm{Ra}-m^2\,r^2-2\,\mathrm{Ra}\,m\,r^2 \\ \frac{8\,\log(r)\,m+r^2\,\log(r)+2\,r^2\,\log(r)\,m^3-r^2-2\,m^3\,r^2}{m} \end{bmatrix}$$

$$\begin{bmatrix} 4\,\frac{r^2}{m} \\ 8+2\,m^2\,r^2+\frac{3}{4}\,\mathrm{Ra}\,m\,r^2 \\ \frac{1}{4}\,\frac{12\,m+8\,r^2+3\,m^3\,r^2}{m} \end{bmatrix}$$

$$\begin{bmatrix} 4\,\frac{r^2(2\,\log(r)-1)}{m} \\ 16\,\log(r)+4\,r^2\,\log(r)\,m^2+\frac{3}{2}\,r^2\,\log(r)\,m\,\mathrm{Ra}-4\,m^2\,r^2-\frac{3}{2}\,\mathrm{Ra}\,m\,r^2 \\ \frac{1}{2}\,\frac{12\,\log(r)\,m+8\,r^2\,\log(r)+3\,r^2\,\log(r)\,m^3-8\,r^2-3\,m^3\,r^2}{m} \end{bmatrix}$$

■

2.8 Some comparison tests

Our prototype of Frobenius' approach follows exactly the method described in Section 2.7. When the associated matrix polynomial $L_0(\lambda)$ has eigenvalues with large partial multiplicities, a lot of differentiations and evaluations have to be performed in order to compute the regular solutions. This leads sometimes to the saturation of the memory machine. Hence, our implementation of Frobenius' method turns out to work only for systems of small size and order. Consequently, we will not give any timings of it here but it could be worth looking for an improved implementation by considering, if possible, the linear systems hidden behind all these differentiations and evaluations.

In the sequel, we give some timings[11] comparing three implementations:

- Bce implements Algorithm **BCE_V2** and it is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`;
- Lfs implements the algorithm of [5] available in the LinearFunctionalSystems package;
- Fos converts System (2.1) into a first-order system of size $n\,\ell$ and then use the implementation of the algorithm given in [21] available in the Isolde package [23].

We have first made tests on first-order systems of the form $\vartheta(y(x))+A(x)\,y(x)=0$, where $A(x)$ is an $n\times n$ matrix polynomial. This case is also of interest since we did not find, in literature, any work adapting Poole's idea to first-order systems. Systems on which we have made our tests have been constructed as follows. We have first chosen a set $\mathcal{S}$ of n' ($n'\leq n$) distinct elements of $\mathbb{Q}$. Then, we have constructed a matrix $A_0\in\mathbb{Q}^{n\times n}$ such that $\sigma(L_0)=\mathcal{S}$, where $L_0(\lambda)=I_n\,\lambda+A_0$. Finally, using the command *RandomMatrix* of the LinearAlgebra package, we have generated 20 matrix polynomials[12] $A(x)\in\mathbb{Q}[x]^{n\times n}$ of degree 10 such that $A(0)=A_0$ and we have run the three algorithms. Timings are presented in Table 2.1: they indicate the average time in seconds taken for computing the general regular solution of the system up to order $\nu=5$ (see Definition 2.6.1). We can remark on these examples that our algorithm is faster than the two others when the matrix A_0 has n distinct eigenvalues which do

[11] Computations were made on a 2.4 GHz Intel Core 2 Duo.
[12] The coefficients having integer values in the range $0,\ldots,50$.

n	$\sigma(L_0)$	BCE	LFS	FOS
2	$\{1,2\}$	0.080	0.052	0.153
4	$\{1,2\}$	0.254	0.192	0.495
8	$\{1,2,3\}$	1.153	1.033	3.218
16	$\{1,2,3,4\}$	14.037	21.044	63.388
2	$\{\frac{1}{2},\frac{1}{3}\}$	0.071	0.083	0.141
4	$\{\frac{1}{2},\frac{1}{3},\ldots,\frac{1}{5}\}$	0.183	0.390	0.451
8	$\{\frac{1}{2},\frac{1}{3},\ldots,\frac{1}{9}\}$	0.775	3.180	2.959
16	$\{\frac{1}{2},\frac{1}{3},\ldots,\frac{1}{17}\}$	11.798	51.248	95.691
2	$\{1\}$	0.063	0.051	0.118
4	$\{1\}$	0.137	0.163	0.271
8	$\{1\}$	0.501	0.673	1.132
16	$\{1\}$	6.186	5.808	20.049

Table 2.1: Timings (in seconds) for $\ell = 1$

ℓ	$\sigma(L_0)$	BCE	LFS	FOS
4	$\{\frac{1}{2},\frac{5}{2},\frac{1}{3}\}$	0.368	1.050	1.354
8	$\{\frac{1}{2},\frac{5}{2},\frac{1}{3}\}$	1.198	9.064	6.484
12	$\{\frac{1}{2},\frac{5}{2},\frac{1}{3}\}$	2.822	41.265	26.573
4	$\{\frac{1}{2},\frac{1}{3}\}$	0.287	1.037	1.310
8	$\{\frac{1}{2},\frac{1}{3}\}$	1.004	8.524	6.812
12	$\{\frac{1}{2},\frac{1}{3}\}$	2.123	34.711	23.634

Table 2.2: Timings (in seconds) for $n = 2$

not differ by integers. It is also faster when $n = 16$ and $\sigma(L_0) = \{1,2,3,4\}$, and when $n = 4, 8$ and the matrix pencil $L_0(\lambda)$ has one eigenvalue.

We have also made comparison tests on higher-order ($\ell \geq 2$) linear differential systems of the form (2.1). We have first made tests for $n = 2$ and $\ell \in \{4, 8, 12\}$. Systems on which we have applied our tests have been constructed as follows. We have first chosen a set $\mathcal{S}$ of $n' \leq n\,\ell = 2\,\ell$ rational numbers. Then, we have constructed a 2×2 upper triangular matrix polynomial $L(\lambda)$ of degree ℓ attained at the diagonal entries and such that $\sigma(L) = \mathcal{S}$. Finally, we have generated 20 systems of the form (2.1) where the coefficient matrices $A_i(x)$, for $i = 0, \ldots, \ell$, are matrix polynomials of degree 10 and the associated matrix polynomial $L_0(\lambda)$ is equal to $L(\lambda)$. Timings are represented in Table 2.2 where the general regular solution is computed up to order $\nu = 5$. Then, we have made other tests on systems with $\ell = 4$ and $n \in \{2, 5, 8\}$. Such systems have been constructed as follows. We have started by choosing a set $\mathcal{S}$ of $n' \leq n\,\ell = 4\,n$ rational numbers, then we have constructed an $n \times n$ diagonal matrix polynomial $D(\lambda)$ such that its diagonal entries are monic polynomials of degree ℓ and $\sigma(D) = \mathcal{S}$. Then, we have set $L(\lambda) = P\,D(\lambda)P^{-1}$, where P is an invertible random matrix whose entries are integers between $0, \ldots, 50$. Finally, we have generated 20 systems of the form (2.1) with coefficient matrices $A_i(x)$ polynomials of degree 10 and whose associated matrix polynomial $L_0(\lambda)$ is equal to $L(\lambda)$. Timings are given in Table 2.3 where the general regular solution is computed up to order $\nu = 5$. Timings in Tables 2.2 and 2.3 show that our algorithm BCE is always faster than LFS and FOS. This can be explained by the fact that our program uses a direct approach while LFS reduces the problem into solving matrix recurrence equations (see [1, 2]) and FOS manipulates a first-order system of bigger size $n\,\ell$ without taking advantage of its companion structure. However, we cannot rely on these tests to claim definitively the performance of our algorithm.

n	$\sigma(L_0)$	BCE	LFS	FOS
2	$\{\frac{1}{2}, \frac{5}{2}, \frac{1}{3}\}$	0.479	1.307	1.648
5	$\{\frac{1}{2}, \frac{5}{2}, \frac{1}{3}\}$	2.396	26.516	39.413
8	$\{\frac{1}{2}, \frac{5}{2}, \frac{1}{3}\}$	7.202	116.363	327.552
2	$\{\frac{1}{2}, \frac{1}{3}\}$	0.291	1.064	1.401
5	$\{\frac{1}{2}, \frac{1}{3}\}$	1.383	17.188	31.158
8	$\{\frac{1}{2}, \frac{1}{3}\}$	5.151	77.646	279.047

Table 2.3: Timings (in seconds) for $\ell = 4$

Chapter 3

Simple Forms of Higher-Order Linear Differential Systems and their Applications in Computing Regular Solutions

Contents

3.1 Introduction

We consider systems of n linear differential equations of order $\ell \geq 1$ of the form

$$\mathcal{L}(x,\vartheta)(y(x)) = A_\ell(x)\,\vartheta^\ell(y(x)) + A_{\ell-1}(x)\,\vartheta^{\ell-1}(y(x)) + \cdots + A_0(x)\,y(x) = 0, \tag{3.1}$$

where $\vartheta = x\,\frac{d}{dx}$, the coefficients $A_i(x)$, for $i = 0,\ldots,\ell$, are $n \times n$ matrices having entries in the ring $\mathbb{K}[[x]]$ and $y(x)$ is an unknown n-dimensional vector. We suppose that $A_\ell(x)$ is a nonzero matrix, but not necessarily invertible and that all the rows of $\mathcal{L}(x,\vartheta)$ are of valuation zero (the valuation with respect to x). We recall that when $A_\ell(0)$ is invertible, such systems are said to be of the first kind and have been investigated in Chapter 2. In this chapter, we are interested in computing the regular formal solutions of System (3.1), *i.e.*, solutions of the form

$$y(x) = x^{\lambda_0} z(x),$$

where $\lambda_0 \in \overline{\mathbb{K}}$ and $z(x) \in \overline{\mathbb{K}}[\log(x)]^n[[x]]$.

With System (3.1), we associate the matrix polynomial

$$\mathcal{L}(0,\lambda) = A_\ell(0)\,\lambda^\ell + A_{\ell-1}(0)\,\lambda^{\ell-1} + \cdots + A_0(0),$$

and we distinguish two cases:

- First case: $\det(\mathcal{L}(0,\lambda)) \neq 0$. In this case, System (3.1) is said to be *simple*. We show that simple systems have regular formal solution spaces of finite dimension over $\overline{\mathbb{K}}$, equal to the degree of $\det(\mathcal{L}(0,\lambda))$ (see Theorem 3.2.1). We also prove that the methods developed in Chapter 2 are still valid to compute regular solutions of simple systems even if $A_\ell(0)$ is not invertible.

- Second case: $\det(\mathcal{L}(0,\lambda)) = 0$. Here, we need to suppose that the leading coefficient matrix $A_\ell(x)$ is invertible in $\mathbb{K}((x))^{n\times n}$ in order to guarantee that System (3.1) has a finite dimensional space of regular formal solutions. Our strategy is to compute a simple linear differential system $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ from which we can get the regular solutions of the original system. Unlike linear differential systems of first-order (cf. [21]), the operator $\overline{\mathcal{L}}(x,\vartheta)$ cannot always be obtained from $\mathcal{L}(x,\vartheta)$ via a transformation of the form $\overline{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$, where $S(x)$ and $T(x)$ are two $n \times n$ invertible matrices with entries in $\mathbb{K}((x))$ (see Theorem 3.4.1 below). For this reason, we are first interested in the existence of a linear substitution $y(x) = T(x)\,z(x)$ with invertible matrix $T(x)$ such that the linear differential system satisfied by $z(x)$ is simple. We develop an algorithm that either decides the existence of such a linear substitution and computes it, or proves that it does not exist. In the latter case, we give a differential variant of the *EG'-algorithm* proposed by Abramov et al. in [4, Section 4] for matrix recurrence equations. Here we need to suppose that the non-simple system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ has polynomial coefficients. This algorithm applies a series of elementary operations to the rows of $\mathcal{L}(x,\vartheta)$ and always yields a simple system from which the regular solutions of the original system can be recovered. Depending on the elementary operations performed, the regular formal solution spaces of these two systems may not be isomorphic; when they are not so, we explain how regular solutions of the non-simple system can be obtained. Finally, we note that we have implemented these algorithms in Maple and we study here their arithmetic complexity.

The remainder of the chapter is organized as follows. Section 3.2 deals with simple systems: we prove that the methods proposed in Chapter 2 can be applied to compute a basis of the regular formal solution space of any simple linear differential system of the form (3.1). In Section 3.3, we review the algorithm proposed in [21] for computing regular formal solutions of first-order simple linear differential systems. We then compare, from an arithmetic complexity point of view, the method developed in Section 2.6 of Chapter 2 with the one consisting in transforming System (3.1) into a first-order linear differential system of size $n\,\ell$ and then using the algorithm of [21]. Section 3.4 introduces our approach to handle non-simple systems and points out the main difference with linear differential systems of first-order. Then, in Section 3.5, we are interested in computing a linear substitution yielding a simple system. We provide both a necessary condition for the existence of such a linear substitution and an algorithm that computes it when it exists. Furthermore, since we deal with systems having formal power series coefficients, we give a bound on the order at which we have to truncate the coefficients of $\mathcal{L}(x,\vartheta)$ in order to get its general regular solution up to a fixed order. Section 3.6 is concerned with the case when the system cannot be reduced to a simple one by means of a linear substitution: we describe a differential variant of the *EG'-algorithm* proposed in [4, Section 4] which always provides a simple system. Then, we explain how to recover the regular formal solutions of the original system from those of the simple one computed by the latter algorithm.

This chapter constitutes the subject of the published paper [18].

Notation. For an $m \times n$ matrix M, we denote by $M(i,j)$ the (i,j)th entry of M and by $M(i,.)$, respectively $M(.,j)$, the ith row, respectively the jth column, of M. For $f \in \mathbb{K}((x)) \setminus$

$\{0\}$, we define the *valuation* $v(f)$ as the order of f at 0. We set $v(0) = +\infty$. For $M \in \mathbb{K}((x))^{m\times n}$, we define $v(M) = \min\{v(M(i,j)), 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}$ and we denote by $\ell c(M)$ the coefficient of $x^{v(M)}$ in M (if $M = 0$, then we set $\ell c(M) = 0$). For a linear differential system of the form (3.1) and $1 \leq i \leq n$, we define

$$\deg_x(\mathcal{L}(x,\vartheta)(i,.)) = \max_{k=0,\ldots,\ell} \deg(A_k(x)(i,.))$$

and similarly,

$$v(\mathcal{L}(x,\vartheta)(i,.)) = \min_{k=0,\ldots,\ell} v(A_k(x)(i,.)).$$

The same definitions hold for the columns of $\mathcal{L}(x,\vartheta)$. Finally, we denote by $\mathbb{A}[\vartheta]$ with $\mathbb{A} = \mathbb{K}[[x]]$, $\mathbb{K}((x))$, etc, the ring of differential operators with coefficients in $\mathbb{A}$, *i.e.*, the set of finite sums $\sum a_i\,\vartheta^i$ with a_i in $\mathbb{A}$ equipped with the addition and the multiplication defined by $\vartheta^i\vartheta^j = \vartheta^{i+j}$, where $i,j \in \mathbb{N}$, and $\vartheta\, a = a\,\vartheta + \vartheta(a)$, where $a \in \mathbb{A}$.

3.2 Regular solutions of simple linear differential systems

In this section, we show that the approach for computing regular solutions of systems of the first kind proposed in Section 2.6 of Chapter 2 can be generalized to handle any system of the form (3.1) whose associated matrix polynomial $\mathcal{L}(0,\lambda)$ is regular. To achieve this, we give new (direct) proofs of the results on which this approach is built.

In the sequel, we follow the terminology used in [21, Def. 2.1] for linear differential systems of first-order:

Definition 3.2.1. *A linear differential system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ (or the matrix differential operator $\mathcal{L}(x,\vartheta)$) of the form (3.1) is said to be* simple *if its associated matrix polynomial $\mathcal{L}(0,\lambda)$ is regular.*

A system of the form (3.1) with invertible matrix $A_\ell(0)$ is necessarily simple, but the converse is not always true.

Example 3.2.1. *Consider the matrix differential operator given by*

$$\mathcal{L}(x,\vartheta) = \begin{pmatrix} 1+x+x^2 & 0 \\ 3\,x^2+x^5 & 0 \end{pmatrix}\vartheta^2 + \begin{pmatrix} 2+5\,x^4 & 3+x^3 \\ 2+x^3+x^4 & 1+2\,x^2+x^4 \end{pmatrix}\vartheta + \begin{pmatrix} 1+5\,x^2 & x+x^2+x^3 \\ 0 & 1+2\,x^2+x^4 \end{pmatrix}.$$

Its associated matrix polynomial

$$\mathcal{L}(0,\lambda) = \begin{pmatrix} \lambda^2+2\,\lambda+1 & 3\,\lambda \\ 2\,\lambda & \lambda+1 \end{pmatrix}$$

is regular since $\det(\mathcal{L}(0,\lambda)) = \lambda^3 - 3\,\lambda^2 + 3\,\lambda + 1$. *Consequently, $\mathcal{L}(x,\vartheta)$ is simple whereas its leading coefficient matrix is not invertible at $x = 0$.* ■

Assume that System (3.1) is simple. We search for regular formal solutions of (3.1) written as

$$y(x) = \sum_{m\geq 0} U_m(t)\, x^{\lambda_0+m}, \tag{3.2}$$

where $\lambda_0 \in \overline{\mathbb{K}}$, $t = \log(x)$, for all $m \geq 0$, $U_m(t) \in \overline{\mathbb{K}}[t]^n$ and $U_0 \neq 0$. We referred to such a solution as a regular solution of exponent λ_0.

Write the coefficient matrices $A_i(x)$, for $i = 0,\ldots,\ell$, of System (3.1) as formal power series

$$A_i(x) = \sum_{i=0}^{\infty} A_{i,j}\, x^j \tag{3.3}$$

with $A_{i,j} \in \mathbb{K}^{n\times n}$. For $j \geq 1$, define the matrix differential operator

$$L_j(\vartheta) = \sum_{i=0}^{\ell} A_{i,j}\, \vartheta^i, \tag{3.4}$$

where the $A_{i,j}$ are given by (3.3). System (3.1) can then be written in the form

$$\sum_{j=1}^{\infty} x^j\, L_j(\vartheta)(y(x)) + \mathcal{L}(0,\vartheta)(y(x)) = 0. \tag{3.5}$$

Plugging (3.2) into (3.5) and using the equality $L_j(\vartheta)\left(x^{\lambda_0+m}\, U_m\right) = x^{\lambda_0+m}\, L_j(\vartheta+\lambda_0+m)(U_m)$, we find that λ_0 and U_0 must satisfy

$$\mathcal{L}(0,\vartheta)\left(x^{\lambda_0}\, U_0\right) = 0, \tag{3.6}$$

and for $m \geq 1$, U_m satisfies

$$\mathcal{L}(0,\vartheta+\lambda_0+m)(U_m) = -\sum_{i=0}^{m-1} L_{m-i}(\vartheta+\lambda_0+i)(U_i). \tag{3.7}$$

Equation (3.6) shows that $x^{\lambda_0}\, U_0$ is a regular solution of the simple system with constant coefficients $\mathcal{L}(0,\vartheta)(y(x)) = 0$. From Lemma 2.4.1 of Chapter 2, which remains valid even if the leading coefficient matrix of $\mathcal{L}(0,\vartheta)(y(x)) = 0$ is not invertible but the system is simple, it follows that λ_0 must be chosen as an eigenvalue of $\mathcal{L}(0,\lambda)$ and U_0 of the form

$$U_0 = v_{k-1} + v_{k-2}\,\frac{\log(x)}{1\,!} + \cdots + v_0\,\frac{\log^{k-1}(x)}{(k-1)\,!},$$

where $v_0, \ldots, v_{k-1}$ form a Jordan chain of length k associated with λ_0.

From Equation (3.7), U_m satisfies a non-homogeneous simple linear differential system with constant coefficients belonging to $\mathbb{K}(\lambda_0)$ and polynomial right-hand side in $t = \log(x)$. In the sequel, we show that Proposition 2.5.1 of Chapter 2 is still valid even if the leading coefficient matrix is not invertible: we only need to suppose that the system is simple.

Proposition 3.2.1. *Consider a non-homogeneous linear differential system with constant coefficients of the form*

$$L(\vartheta)(y(x)) = \mathcal{A}_\ell\, \vartheta^\ell(y(x)) + \mathcal{A}_{\ell-1}\, \vartheta^{\ell-1}(y(x)) + \cdots + \mathcal{A}_0\, y(x) = \phi(t), \tag{3.8}$$

where for $i = 0, \ldots, \ell$, $\mathcal{A}_i \in \mathbb{F}^{n\times n}$ ($\mathbb{K} \subseteq \mathbb{F} \subseteq \mathbb{C}$), $t = \log(x)$ and $\phi(t) \in \mathbb{F}[t]^n$ of degree d. If the system is simple, i.e., $\det(L(\lambda)) \neq 0$, *then it admits at least a polynomial solution in t of degree p with coefficients in $\mathbb{F}$ such that*

$$\begin{cases} p = d & \text{if} \quad 0 \notin \sigma(L), \\ d \leq p \leq d + \max\{\kappa_j,\ j = 1, \ldots, m_g(0)\} & \text{if} \quad 0 \in \sigma(L), \end{cases}$$

where $\kappa_1, \ldots, \kappa_{m_g(0)}$ denote the partial multiplicities of $L(\lambda)$ at the eigenvalue 0.

The proof given in Chapter 2 uses explicitly the hypothesis "$\mathcal{A}_\ell$ is invertible" to convert System (3.8) into a first-order one. Consequently, we provide here a new proof.

Proof. Let $S(\lambda) = \mathrm{diag}(a_1(\lambda), \ldots, a_n(\lambda))$, where the a_i's are monic polynomials in λ with coefficients in $\mathbb{F}$, be the Smith normal form of $L(\lambda)$ (see Theorem 1.2.1 of Chapter 1). Then, there exist two unimodular matrix polynomials $E(\lambda)$ and $F(\lambda)$ in $\mathbb{F}[\lambda]^{n\times n}$ such that $E(\lambda)\,L(\lambda) = S(\lambda)\,F(\lambda)$. Put $\psi(t) = E(\vartheta)(\phi(t)) \in \mathbb{F}[t]^n$ and $z(x) = F(\vartheta)(y(x))$. System (3.8) is then equivalent to the system $S(\vartheta)(z(x)) = \psi(t)$ and y is a polynomial solution in t of (3.8) of degree p if and only if z is a polynomial solution in t of $S(\vartheta)(z(x)) = \psi(t)$ of the same degree p since $F(\lambda)$ is unimodular. Furthermore, since $E(\lambda)$ is unimodular, we have $\deg(\psi(t)) = \deg(\phi(t)) = d$. Let $z = (z_1, \ldots, z_n)^T$ and $\psi = (\psi_1, \ldots, \psi_n)^T$. We show now that the scalar linear differential equation $a_i(\vartheta)(z_i(x)) = \psi_i(t)$ has a polynomial solution in $t = \log(x)$ and give a bound on its degree. Write z_i as a polynomial in $t = \log(x)$, plug it into $a_i(\vartheta)(z_i(x)) = \psi_i(t)$ and identify the coefficients of the powers of t. Two cases must be considered: if $a_i(0) \neq 0$, then the coefficients of z_i are uniquely determined and the degree in t of z_i is equal to that of ψ_i. Otherwise, $a_i(\lambda)$ is of the form $\lambda^{\kappa_i} b_i(\lambda)$ where $b_i(0) \neq 0$ and $\kappa_i \in \mathbb{N}^*$ is one of the partial multiplicities of $L(\lambda)$ at 0. In this case, the scalar differential equation $a_i(\vartheta)(z_i(x)) = \psi_i(t)$ admits a polynomial solution in $t = \log(x)$ of degree equal to $\deg(\psi_i(t)) + \kappa_i \leq d + \kappa_i$, which ends the proof. □

We also stress that Corollary 2.5.1 of Chapter 2 which gives the general polynomial solution in $t = \log(x)$ and Algorithm **GenPolSol_NHS** which computes this general solution remain valid for any simple linear differential system of the form (3.8).

Thus, for all $m \geq 1$, System (3.7) always admits a polynomial solution $U_m(t)$. Hence, every regular solution $x^{\lambda_0} U_0$ of System (3.6) can be extended to a regular formal solution of exponent λ_0 of System (3.1). As the dimension of the regular formal solution space of system $\mathcal{L}(0, \vartheta)(y(x)) = 0$ is equal to $\deg(\det(\mathcal{L}(0, \lambda)))$, see [50, Th. S1.6], this provides $\deg(\det(\mathcal{L}(0, \lambda)))$ linearly independent regular formal solutions of System (3.1). We prove in the next theorem that this is exactly the dimension of the regular formal solution space of System (3.1).

Theorem 3.2.1. *The dimension of the regular formal solution space of a simple linear differential system of the form (3.1) is equal to* $\deg(\det(\mathcal{L}(0, \lambda)))$.

Proof. Let $\mathcal{V}$ denote the $\overline{\mathbb{K}}$-vector space spanned by the $\deg(\det(\mathcal{L}(0, \lambda)))$ linearly independent regular formal solutions computed by the method described above. Suppose that the dimension of the regular formal solution space of the simple system $\mathcal{L}(x, \vartheta)(y(x)) = 0$ is greater than $\deg(\det(\mathcal{L}(0, \lambda)))$. Then there exists a regular formal solution $y(x)$ of (3.1) which is not in $\mathcal{V}$. Let $y(x) = \sum_{i=0}^{\infty} U_i\, x^{\lambda_0 + i}$ be such a solution with $\lambda_0 \in \overline{\mathbb{K}}$, $U_i \in \overline{\mathbb{K}}[\log(x)]^n$ and $U_0 \neq 0$. We assume, without loss of generality, that the real part of λ_0 is maximal among those of the exponents of the regular formal solutions of (3.1) which do not belong to $\mathcal{V}$. Since $y(x)$ is a regular solution of $\mathcal{L}(x, \vartheta)(y(x)) = 0$, we know from the discussion above that necessarily $x^{\lambda_0} U_0$ satisfies $\mathcal{L}(0, \vartheta)\left(x^{\lambda_0} U_0\right) = 0$. Consequently, there exists a regular formal solution $z(x) = \sum_{i=0}^{\infty} V_i\, x^{\lambda_0 + i} \in \mathcal{V}$ of (3.1) with $V_0 = U_0$. We point out that $z(x)$ is a linear combination of the regular formal solutions of exponent λ_0 belonging to $\mathcal{V}$. Now, $y(x) - z(x)$ is a nonzero regular formal solution of (3.1) not belonging to $\mathcal{V}$. Moreover, since $V_0 = U_0$, we have $y(x) - z(x) = x^{\lambda_0 + j} \sum_{i=0}^{\infty} W_i\, x^i$ with $j \in \mathbb{N}^*$ and $W_0 \neq 0$. This is in contradiction with the fact that $\Re(\lambda_0)$ is maximal. □

We have thus proven that Algorithm **BCE_V1** developed in Subsection 2.6.2 of Chapter 2 can be applied to compute a basis of the regular formal solution space of any linear differential system of the form (3.1) assuming only that it is simple, *i.e.*, $\det(\mathcal{L}(0, \lambda)) \neq 0$. Similarly, the other variant **BCE_V2** developed in Subsection 2.6.3 of Chapter 2 works as well on simple linear differential systems of the form (3.1) and produces its general regular solution.

Now, concerning our generalization of the Frobenius method described in Subsection 2.7 of Chapter 2, it remains correct for any simple linear differential system of the form (3.1) not necessarily of the first kind. This follows from the fact that to achieve the generalization, we have only

used the regularity of the matrix polynomial $\mathcal{L}(0,\lambda)$ (which was denoted by $L_0(\lambda)$ in Chapter 2).

Finally, we note that Theorem 3.2.1 above allows us to easily detect whether a simple system of the form (3.1) with $A_\ell(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ has a regular singularity at $x = 0$ or not, as stated in the following corollary.

Corollary 3.2.1. *A simple linear differential system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ of the form (3.1) with invertible leading coefficient matrix $A_\ell(x)$ in $\mathbb{K}((x))^{n\times n}$ has a regular singularity at the point $x = 0$ if and only if it is of the first kind.*

Proof. The system has a regular singularity at $x = 0$ if and only if all its formal solutions are regular solutions, namely, if and only if the dimension of its regular formal solution space is equal to $n\,\ell$. According to Theorem 3.2.1, this occurs if and only if $\deg(\det(\mathcal{L}(0,\lambda))) = n\,\ell$, *i.e.*, $A_\ell(0)$ is invertible (see Section 1.1 of Chapter 1). □

3.3 Transformation to a first-order linear differential system: Algorithm of Barkatou-Pflügel

Another approach for computing regular formal solutions of a linear differential system of the form (3.1) consists in converting it into a first-order linear differential system and then using one of the algorithms dedicated to the first-order case: see [8, 21, 37, 56].

In this section, we sketch the algorithm proposed in [21][1] for computing regular formal solutions of simple linear differential systems of first-order and we study its arithmetic complexity. Then, we compare, from an arithmetic complexity point of view, Algorithm **BCE_V2** which handles directly a simple linear differential system of the form (3.1) to the approach consisting in transforming (3.1) into the first-order linear differential system of size $n\,\ell$

$$D(x)\,\vartheta(Y(x)) - N(x)\,Y(x) = 0,$$

where

$$D(x) = \begin{pmatrix} I_{n\,(\ell-1)} & 0 \\ 0 & A_\ell(x) \end{pmatrix}, \quad N(x) = \begin{pmatrix} 0 & I_n & 0 & \dots & 0 \\ 0 & 0 & I_n & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \dots & I_n \\ -A_0(x) & -A_1(x) & \dots & \dots & -A_{\ell-1}(x) \end{pmatrix} \tag{3.9}$$

and $Y(x) = (y(x)^T, \vartheta(y(x))^T, \dots, \vartheta^{\ell-1}(y(x))^T)^T$, and then applying the algorithm of [21]. Note that the resulting first-order system is also simple since $D(0)\,\lambda - N(0)$ is a linearization of $\mathcal{L}(0,\lambda)$ (see Section 1.3 of Chapter 1).

Consider a simple linear differential system of first-order of the form

$$\mathcal{D}(y(x)) = D(x)\,\vartheta(y(x)) - N(x)\,y(x) = 0, \tag{3.10}$$

where $D(x) = \sum_{j\geq 0} D_j\,x^j \in \mathbb{K}[[x]]^{n\times n}$ and $N(x) = \sum_{j\geq 0} N_j\,x^j \in \mathbb{K}[[x]]^{n\times n}$.

In [21], the authors look for regular solutions of System (3.10) written in the form

$$y(x) = x^{\lambda_0}\left(h_s(x) + \log(x)\,h_{s-1}(x) + \cdots + \frac{\log^{s-1}(x)}{(s-1)!}\,h_1(x)\right),$$

[1] This algorithm has been implemented by the authors of [21] in Maple and can be found in the package Isolde [23].

where $s \in \mathbb{N}^*$, $\lambda_0 \in \overline{\mathbb{K}}$ and for $k = 1, \ldots, s$, $h_k(x) \in \overline{\mathbb{K}}[[x]]^n$ such that $h_1(x) \neq 0$. We know that λ_0 must be chosen as an eigenvalue of the matrix pencil $D_0\,\lambda - N_0$. Thus, by gathering the eigenvalues of $D_0\,\lambda - N_0$ into pairwise disjoint sets $\sigma_1, \ldots, \sigma_r$ such that two eigenvalues belonging to two different sets σ_i and σ_j do not differ by integers, one needs to compute the general regular solution associated with each set σ_i (see page 54). For this, let $\lambda_i \in \sigma_i$ such that $\Re(\lambda_i) = \min_{\lambda_0 \in \sigma_i} \Re(\lambda_0)$ and let

$$y_i(x) = x^{\lambda_i} \sum_{k=1}^{s_i} \frac{\log^{s_i-k}(x)}{(s_i-k)!}\, h_{i,k}(x), \tag{3.11}$$

where the $h_{i,k}$'s are vectors of formal power series and $h_{i,1}(x) \neq 0$, be the general regular solution associated with σ_i to be computed. Let

$$m_i = \sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0). \tag{3.12}$$

From Equation (2.28) in Chapter 2, we can deduce that the degree s_i is bounded by the sum of the algebraic multiplicities of the eigenvalues belonging to σ_i, *i.e.*, $s_i \leq m_i$. Note that the number of arbitrary constants appearing in $h_{i,k}$ for $k = 1, \ldots, s_i$ is equal to m_i. According to [21, Lemma 3.1], $y_i(x)$ is a solution of $\mathcal{D}(y(x)) = 0$ if and only if

$$\begin{cases} \mathcal{D}_i(h_{i,1}(x)) = 0, \\ \mathcal{D}_i(h_{i,k}(x)) = -D(x)\, h_{i,k-1}(x) \quad \text{for} \quad 2 \leq k \leq s_i, \end{cases}$$

where $\mathcal{D}_i = D(x)\,\vartheta - (N(x) - D(x)\,\lambda_i) \in \mathbb{K}(\lambda_i)[[x]][\vartheta]^{n\times n}$. Consequently, computing the regular solution $y_i(x)$ of the form (3.11) is reduced to finding the general formal power series solution of at most m_i linear differential system of first-order of the form

$$\mathcal{D}_i(y(x)) = (D(x)\,\vartheta - N(x) + D(x)\,\lambda_i)(y(x)) = b(x), \tag{3.13}$$

where b is an n-dimensional vector of formal power series depending linearly on parameters.

Let us now explain how the authors of [21] proceed to compute the general formal power series solution up to order $\nu \in \mathbb{N}$ of System (3.13). Solving System (3.13) means finding the set of all parameters appearing in the right-hand side b for which the system is consistent and computing its general power series solution. Let $p_1, \ldots, p_l$ denote the parameters appearing in b and write

$$b(x) = \ell c(b)\, x^{\delta} + \cdots,$$

where $\delta \in \mathbb{N}$, $\ell c(b)$ depends linearly on some of the p_i's and the dots stand for terms of valuation greater than δ. Write

$$y(x) = c\, x^{\mu} + z(x)$$

with $\mu \in \mathbb{N}$ such that $\mu \leq \nu$ and c is a nonzero vector. The problem is to find an integer $\mu \leq \nu$ and a nonzero vector c such that $\mathcal{D}_i(y(x)) = b(x)$ and $v(z) > \mu$. Plugging $y(x)$ into (3.13), one gets

$$\mathcal{D}_i(z(x)) = b(x) - \mathcal{D}_i(c\, x^{\mu}) = \left[\ell c(b)\, x^{\delta} + \cdots\right] - \left[(D_0\,\mu - N_0 + D_0\,\lambda_i)\, c\, x^{\mu} + \cdots\right]. \tag{3.14}$$

Since $v(\mathcal{D}_i(z)) \geq v(z) > \mu$ (see [21, Lemma 2.1]), a necessary condition for the existence of a couple (μ, c) is that the valuation of the right-hand side of (3.14) must be greater than μ. This occurs only if one chooses $\mu < \delta$ such that $\det(D_0\,\mu - N_0 + D_0\,\lambda_i) = 0$ and c such that $(D_0\,\mu - N_0 + D_0\,\lambda_i)\, c = 0$, or $\mu = \delta$ and c satisfying $(D_0\,\mu - N_0 + D_0\,\lambda_i)\, c = \ell c(b)$. So let $\mathcal{R} = \sigma(D_0\,\lambda - N_0 + D_0\,\lambda_i) \cap \{0, 1, \ldots, \nu\}$ and let $\mathcal{P} = \{p_1, \ldots, p_l\}$ denote the set of parameters appearing in the right-hand side of (3.14). There are several cases:

- If $\delta > \nu$ and $\mathcal{R} = \emptyset$, then there is no possible couple (μ, c).
- If there is a $\mu_0 \in \mathcal{R}$ such that $\mu_0 < \delta$, then take $\mu = \mu_0$ and c as the general solution of $(D_0\,\mu - N_0 + D_0\,\lambda_i)\,c = 0$ (c depends then on $m_g(\lambda_i + \mu)$ parameters[2]).
- If $\forall\,\mu_0 \in \mathcal{R}$, one has $\mu_0 \geq \delta$, then two cases have to be distinguished: if one can determine some elements of $\mathcal{P}$ in such a way that $\ell c(b)$ belongs to the range of the matrix $D_0\,\delta - N_0 + D_0\,\lambda_i$, then take $\mu = \delta$ and c as the general solution of $(D_0\,\mu - N_0 + D_0\,\lambda_i)\,c = \ell c(b)$. Otherwise, *i.e.*, $\ell c(b)$ does not belong to the range of $D_0\,\delta - N_0 + D_0\,\lambda_i$ for all choices of the p_i's, then there is no possible couple (μ, c).

After having found the couple (μ, c), one performs the substitution $y(x) = c\,x^{\mu} + z(x)$ into (3.13). This provides a linear differential system satisfied by $z(x)$ of the form

$$\mathcal{D}_i(z(x)) = b(x) - \mathcal{D}_i(c\,x^{\mu}) \tag{3.15}$$

and the process is iterated by updating the sets $\mathcal{R}$ and $\mathcal{P}$: $\mathcal{R} = \sigma(D_0\,\lambda - N_0 + D_0\,\lambda_i) \cap (\{0, 1, \ldots, \nu\} \setminus \{\mu\})$ and $\mathcal{P}$ contains the parameters appearing in the right-hand side of (3.15) (for more details, see [21, pages 575-576]). Note that the cardinality of the set $\mathcal{P}$ is always bounded by $l + \sum_{\lambda_0 \in \sigma_i} m_g(\lambda_0) \leq l + n$ (here $m_g(\lambda_0)$ denote the geometric multiplicity of λ_0 as an eigenvalue of the matrix pencil $D_0\,\lambda - N_0$). Thus, the general formal power series solution $y(x)$ of System (3.13) obtained by proceeding as explained above depends on the p_i's and on at most $\sum_{\lambda_0 \in \sigma_i} m_g(\lambda_0)$ new parameters.

Lemma 3.3.1. *With the previous notation, computing the general formal power series solution of System (3.13) up to order ν can be done in at most $O\big(n^2\,(l+n)\,\nu^2\,d_{\lambda_i}\big)$ arithmetic operations in $\mathbb{K}$, where l denotes the number of parameters in the right-hand side b of (3.13) and d_{λ_i} denotes the degree of the extension $\mathbb{K}(\lambda_i)$ over $\mathbb{K}$.*

Proof. Since we are interested in computing the general power series solution of System (3.13) up to order ν so we can suppose that the right-hand side b of System (3.13) is truncated at order ν. Let now study the cost of solving the system up to order ν. First of all, we compute a LU decomposition of each matrix $D_0\,\mu - N_0 + D_0\,\lambda_i$ for $\mu = 0, \ldots, \nu$. In this way, we reduce the problem of solving linear systems of the form $(D_0\,\mu - N_0 + D_0\,\lambda_i)\,c = w$ to that of solving two linear systems with triangular matrices in $\mathbb{K}(\lambda_i)^{n \times n}$. This can be done in at most $O\big(n^3\,\nu\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. After this, solving System (3.13) can be done in at most $O\big(n^2\,(l+n)\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Indeed, the computation of formal power series solutions of System (3.13) up to order ν is a recursive procedure repeated at most $\nu + 1$ times and composed of three essential steps:

1. computing c as the general solution of a linear algebraic system of the form

$$(D_0\,\mu - N_0 + D_0\,\lambda_i)\,c = w,$$

   where w depends on at most $l + n$ parameters;

2. updating the parameters in the right-hand side b of System (3.13) and the so-far-computed terms when some conditions on the parameters of w are imposed in order to the system be consistent;

3. computing the right-hand side up to order ν of the linear differential system satisfied by z obtained after substituting $y = c\,x^{\mu} + z$.

[2] $m_g(\lambda_i + \mu)$ denotes the geometric multiplicity of $\lambda_i + \mu$ as an eigenvalue of the matrix pencil $D_0\,\lambda - N_0$.

For the computation of c, write $D_0\,\mu - N_0 + D_0\,\lambda_i = P\,L\,U$ where P is a permutation matrix, L a lower triangular matrix having 1 on its diagonal and U is an upper triangular matrix whose last $n-r$ rows are zero (r being the rank of matrix $D_0\,\mu - N_0 + D_0\,\lambda_i$). Thus, solving $(D_0\,\mu - N_0 + D_0\,\lambda_i)\,c = w$ is reduced to solving $L\,c' = P^T\,w$ and then $U\,c = c'$. Since L is a lower invertible triangular matrix and w depends on at most $l+n$ parameters, solving $L\,c' = P^T\,w$ can be done in at most $O\big(n^2\,(l+n)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Now, to solve system $U\,c = c'$, we have to ensure that the last $n-r$ components of c' are zero (c' depends on the parameters appearing in w so on at most $l+n$ parameters). This is equivalent to solving a system of $n-r$ linear algebraic equations in at most $l+n$ variables. So it can be performed in at most $O\big((n-r)^2\,(l+n)\,d_{\lambda_i}\big) = O\big(n^2\,(l+n)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. This gives R parameters ($R \leq n-r$) written as linear combinations of the other parameters which are of number bounded by $l+n-R$. Thus, updating one component of vector c' costs at most $O(R\,(l+n-R)\,d_{\lambda_i}) = O(n\,(l+n)\,d_{\lambda_i})$ operations in $\mathbb{K}$. So the cost of updating vector c' is bounded by $O\big(n^2\,(l+n)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Now vector c' depends on at most $l+n-R$ parameters, so solving system $U\,c = c'$ can be done in at most $O\big(n^2\,(l+n)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Hence, the cost of computing c (Step 1 above) is bounded by $O\big(n^2\,(l+n)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$.

In Step 2, we have at most $\nu+\mu$ vectors with n components to be updated. So, the cost of Step 2 is bounded by $O\big(n^2\,(l+n)\,(\nu+\mu)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$.

In Step 3, the cost of computing of the new right-hand side up to order ν is reduced to that of computing $\mathcal{D}_i(c\,x^\mu)$ truncated at order ν, *i.e.*, $\sum_{j=0}^{\nu-\mu}(D_j\,\mu - N_j + D_j\,\lambda_i)\,c\,x^{\mu+j}$. So we have $\nu-\mu+1$ products of an $n\times n$ matrix with coefficients in $\mathbb{K}(\lambda_i)$ by a n-dimensional vector which depends on at most $l+n$ parameters. Thus, Step 3 costs at most $O\big(n^2\,(l+n)\,(\nu-\mu)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$.

Consequently, one call of the procedure costs at most $O\big(n^2\,(l+n)\,(\nu+\mu)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. As μ goes from 0 to ν, computing the general formal power series solution of System (3.13) up to order ν can be done in at most $O\big(n^2\,(l+n)\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. □

Remark 3.3.1. Suppose that $h_{i,1},\ldots,h_{i,k-1}$ (with $k-1<n$) appearing in (3.11) have been already computed and that $h_{i,1}\neq 0$. The vector defined by

$$y(x) = x^{\lambda_i}\left(h_{i,k-1} + \log(x)\,h_{i,k-2} + \cdots + \frac{\log^{k-2}(x)}{(k-2)!}\,h_{i,1}(x)\right)$$

forms a regular solution of System (3.10) (cf. [21], page 578) and hence one can deduce that the number of arbitrary constants appearing in $h_{i,1},\ldots,h_{i,k-1}$ does not exceed n. Thus, according to Lemma 3.3.1, computing $h_{i,k}(x)$ as the general power series solution up to order ν of System

$$\mathcal{D}_i(y(x)) = -D(x)\,h_{i,k-1}(x)$$

can be performed using at most $O\big(n^3\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$ since l, the number of parameters in $D(x)\,h_{i,k-1}(x)$, is bounded by n.

The algorithm of [21] can be sketched as follows:

ALGORITHM **BP**

INPUT: An integer $\nu \in \mathbb{N}$ and the coefficient matrices $D(x)$ and $N(x)$ truncated at order ν of the simple first-order linear differential system (3.10).
OUTPUT: The general regular solution of (3.10) up to order ν.

1. Compute $\sigma(D(0)\,\lambda - N(0))$ and gather the eigenvalues that differ by integers into sets $\sigma_1, \ldots, \sigma_r$;
2. **For** i from 1 to r **do**
   2.1. Let $\lambda_i \in \sigma_i$ be such that $\Re(\lambda_i) = \min_{\lambda_0 \in \sigma_i} \Re(\lambda_0)$ and let $\mathcal{D}_i = D\,\vartheta - (N - D\,\lambda_i)$;
   2.2. Compute the general formal power series solution of $\mathcal{D}_i(h_{i,1}) = 0$ up to order ν;
   Set $k = 1$;
   2.3. **While** $h_{i,1} \neq 0$ **do**
   2.3.(a). Let $k = k + 1$;
   2.3.(b). Compute the general formal power series solution of system $\mathcal{D}_i(h_{i,k}) = -D\,h_{i,k-1}$ up to order ν;
   2.3.(c). Update the parameters in $h_{i,1}, \ldots, h_{i,k-1}$;
   end do;
   2.4. Set $s_i = k - 1$ and $y_i = x^{\lambda_i} \sum_{j=1}^{s_i} \dfrac{\log^{s_i - j}(x)}{(s_i - j)!}\, h_{i,j}(x)$;
   end do;
3. **Return** $y = \sum_{i=1}^{r} y_i$;

Proposition 3.3.1. *Algorithm* **BP** *computes the general regular solution of System (3.10) up to order ν using at most $O\big(n^5\,\nu + n^4\,\nu^2\big)$ arithmetic operations in $\mathbb{K}$.*

Proof. Let us first determine the cost of computing the general regular solution associated with σ_i up to order ν. From Lemma 3.3.1, Step 2.2 can be done using at most $O\big(n^3\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. In Step 2.3.(b), computing the right-hand side $D\,h_{i,k-1}$ truncated at order ν can be done using at most $O\big(n^3\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$ since the number of matrix-vector products with entries in $\mathbb{K}(\lambda_i)$ is $\frac{1}{2}(\nu+1)\,(\nu+2)$ and $h_{i,k-1}$ depends on at most n parameters. Now from Lemma 3.3.1, solving the system in Step 2.3.(b) can be done using at most $O\big(n^3\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Hence, the cost of Step 2.3.(b) is bounded by $O\big(n^3\,\nu^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. In Step 2.3.(c), we have $(k-1)\,(\nu+1)$ n-dimensional vector in $\mathbb{K}(\lambda_i)$ depending on parameters, whose number does not exceed n, to be updated. Updating one component of these vectors can be done in at most $O\big(n^2\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Hence the cost of Step 2.3.(c) is bounded by $O\big(n^3\,k\,\nu\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Consequently, the kth passage in the **While** loop costs at most $O\big((n^3\,k\,\nu + n^3\,\nu^2)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Since k goes from 1 to s_i which is bounded by m_i given by (3.12), computing the general regular solution up to order ν associated with σ_i can be done in at most $O\big((n^3\,m_i^2\,\nu + n^3\,m_i\,\nu^2)\,d_{\lambda_i}\big)$ operations in $\mathbb{K}$. Now, we have $\sum_{i=1}^{r} m_i\,d_{\lambda_i} = \sum_{i=1}^{r} \sum_{\lambda_0 \in \sigma_i} m_a(\lambda_0)\,d_{\lambda_i} \leq n$ and $\sum_{i=1}^{r} m_i^2\,d_{\lambda_i} \leq n^2$ so Algorithm **BP** returns the general regular solution of System (3.10) up to order ν after at most $O\big(n^5\,\nu + n^4\,\nu^2\big)$ arithmetic operations in $\mathbb{K}$. □

We have now two different approaches to compute the general regular solution up to order ν of a given simple linear differential system (3.1) of order ℓ and size n:

- First approach: application of Algorithm **BCE_V2** directly to System (3.1). This algorithm uses $O\big(n^4\,\ell^3\,\nu^2 + n^6\,\ell^4\big)$ arithmetic operations in $\mathbb{K}$ (see Proposition 2.6.2 of Chapter 2).

- Second approach: conversion of System (3.1) into the first-order linear differential system $D(x)\,\vartheta(Y(x)) - N(x)\,Y(x) = 0$, where $D(x)$ and $N(x)$ are given by (3.9), then application of Algorithm **BP** above. This can be done using at most $O\big(n^5\,\ell^5\,\nu + n^4\,\ell^4\,\nu^2\big)$ arithmetic operations in $\mathbb{K}$ since the resulting first-order system is of size $n\,\ell$.

3.4 Non-simple systems

In the remainder of the chapter, we consider systems of the form (3.1) which are not simple, *i.e.*, for which $\det(\mathcal{L}(0,\lambda)) = 0$. In order to compute the regular formal solution space of a non-simple system, we propose to compute another differential system which is simple and from which one can get the solutions of the non-simple one.

From now on, we will assume that the leading coefficient matrix $A_\ell(x)$ **is invertible in** $\mathbb{K}((x))^{n\times n}$: this guarantees that the regular formal solution space of System (3.1) is of finite dimension since it can be converted into a first-order system of the form $\vartheta(Y(x)) = C(x)\,Y(x)$ where $C(x) = D(x)^{-1}N(x) \in \mathbb{K}((x))^{n\,\ell\times n\,\ell}$ with $D(x)$ and $N(x)$ given by (3.9). The invertibility of $A_\ell(x)$ also allows us to suppose, without loss of generality, that $A_\ell(x) \in \mathbb{K}[x]^{n\times n}$: indeed, let $\widetilde{\mathcal{L}}(x,\vartheta) = A_\ell^{-1}(x)\,\mathcal{L}(x,\vartheta)$ and let, for $i = 1,\ldots,n$, $\alpha_i = \min\Big\{0, v\Big(\widetilde{\mathcal{L}}(x,\vartheta)(i,.)\Big)\Big\}$ and $S(x) = \mathrm{diag}(x^{-\alpha_1},\ldots,x^{-\alpha_n}) \in \mathbb{K}[x]^{n\times n}$. Multiplying $\widetilde{\mathcal{L}}(x,\vartheta)$ on the left by $S(x)$, we get a new matrix differential operator with coefficient matrices in $\mathbb{K}[[x]]^{n\times n}$ and having $S(x) \in \mathbb{K}[x]^{n\times n}$ as a leading coefficient matrix.

The problem of computing a simple system, from which one can recover the solutions of the non-simple system $\mathcal{L}(x,\vartheta)(y(x)) = 0$, has been already treated in [21] for the case of first-order systems, *i.e.*, when $\ell = 1$: the authors have shown that, using the super-reduction algorithm [24, 25, 58], any non-simple matrix differential operator $\mathcal{L}(x,\vartheta)$ of first-order can be reduced to an equivalent simple one of first-order as well. More precisely, they have shown that there exist two matrices $S(x)$ and $T(x)$ in $\mathrm{GL}_n(\mathbb{K}((x)))$ such that the operator $\overline{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x) \in \mathbb{K}[[x]][\vartheta]^{n\times n}$ is simple; this implies that the two regular formal solution spaces of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ and $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ are isomorphic since $y(x)$ and $z(x)$ are related by $y(x) = T(x)\,z(x)$ with $T(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$. In the next chapter, we will develop a new algorithm computing an equivalent simple operator when $\ell = 1$ without making use of the super-reduction. However, when $\ell \geq 2$, it is not always possible to reduce a non-simple system of the form (3.1) to a simple one using only algebraic transformations $S(x)$ and $T(x)$ in $\mathrm{GL}_n(\mathbb{K}((x)))$ as we will show in the sequel.

Proposition 3.4.1. *Consider a non-simple matrix differential operator of the form*

$$\mathcal{L}(x,\vartheta) = A_2(x)\,\vartheta^2 + A_1(x)\,\vartheta + A_0(x)$$

with $A_2(x)$, $A_1(x)$ and $A_0(x)$ in $\mathbb{K}[[x]]^{2\times 2}$. Assume that its associated matrix polynomial $\mathcal{L}(0,\lambda)$ is equal to

$$\mathcal{L}(0,\lambda) = A_2(0)\,\lambda^2 + A_1(0)\,\lambda + A_0(0) = \begin{pmatrix} \lambda^2 & \lambda \\ \lambda & 1 \end{pmatrix}.$$

For any nonzero matrices $S(x)$ and $T(x)$ in $\mathbb{K}((x))^{2\times 2}$ such that $\widetilde{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ has formal power series coefficients and such that $\widetilde{\mathcal{L}}(0,\lambda) \neq 0$, the matrix differential operator $\widetilde{\mathcal{L}}(x,\vartheta)$ is non-simple.

Proof. Write

$$T(x) = T_\alpha\,x^\alpha + \cdots = \begin{pmatrix} a & b \\ c & d \end{pmatrix} x^\alpha + \cdots \quad \text{and} \quad S(x) = S_\beta\,x^\beta + \cdots,$$

where α, $\beta \in \mathbb{Z}$, T_α, S_β are two nonzero matrices of $\mathbb{K}^{2\times 2}$ and the dots stand for terms of higher valuation. The coefficient matrices of the operator $\widetilde{\mathcal{L}}(x,\vartheta) = \widetilde{A}_2(x)\vartheta^2 + \widetilde{A}_1(x)\vartheta + \widetilde{A}_0(x)$ defined by $\widetilde{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ are given by

$$\widetilde{A}_2(x) = S(x)\,A_2(x)\,T(x) = S_\beta \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix} x^{\alpha+\beta} + \cdots,$$

$$\widetilde{A}_1(x) = S(x)\,(2\,A_2(x)\,\vartheta(T(x)) + A_1(x)\,T(x)) = S_\beta \begin{pmatrix} c+2\,\alpha\,a & d+2\,\alpha\,b \\ a & b \end{pmatrix} x^{\alpha+\beta} + \cdots,$$

$$\widetilde{A}_0(x) = S(x)(A_2(x)\vartheta^2(T(x)) + A_1(x)\vartheta(T(x)) + A_0(x)T(x)) = S_\beta \begin{pmatrix} \alpha^2\,a+\alpha\,c & \alpha^2\,b+\alpha\,d \\ c+\alpha\,a & d+\alpha\,b \end{pmatrix} x^{\alpha+\beta} + \cdots,$$

where the dots stand for terms of valuation greater than $\alpha+\beta$. Consider now the matrix polynomial

$$M(\lambda) = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix} \lambda^2 + \begin{pmatrix} c+2\,\alpha\,a & d+2\,\alpha\,b \\ a & b \end{pmatrix} \lambda + \begin{pmatrix} \alpha^2\,a+\alpha\,c & \alpha^2\,b+\alpha\,d \\ c+\alpha\,a & d+\alpha\,b \end{pmatrix}.$$

One can easily check that it is a singular and none of its rows is zero since T_α is assumed to be a nonzero constant matrix. Moreover, its left nullspace is spanned by the row vector $(-1 \quad \lambda+\alpha)$. If $\alpha+\beta < 0$, then the coefficient matrices of $x^{\alpha+\beta}$ in $\widetilde{A}_2(x)$, $\widetilde{A}_1(x)$ and $\widetilde{A}_0(x)$ are necessarily equal to zero since $\widetilde{\mathcal{L}}(x,\vartheta) \in \mathbb{K}[[x]][\vartheta]^{2\times 2}$. This implies that $S_\beta\,M(\lambda) = 0$ with S_β a nonzero constant matrix which is impossible since there is no nonzero constant vectors in the left nullspace of $M(\lambda)$. Consequently, we have $\alpha+\beta = 0$ (since we suppose that $\widetilde{\mathcal{L}}(0,\lambda) \neq 0$) and $\widetilde{\mathcal{L}}(0,\lambda)$ is equal to $S_\beta\,M(\lambda)$ and hence it is singular. □

The above proposition can be applied to any 2×2 non-simple matrix differential operator of second-order whose associated matrix polynomial has no nonzero constant vectors in its left and right minimal bases. This can be shown using the result of [90, Th. 4.47] which can be stated as follows:

Lemma 3.4.1. *A matrix polynomial $L(\lambda) \in \mathbb{K}[\lambda]^{2\times 2}$ is singular if and only if one of the following three cases happens:*

- *$L(\lambda)$ has a nonzero constant vector in its right minimal basis;*
- *$L(\lambda)$ has a nonzero constant vector in its left minimal basis;*
- *There exist two nonsingular matrices S_0 and T_0 of $\mathbb{K}^{2\times 2}$ such that*

$$S_0\,L(\lambda)\,T_0 = \begin{pmatrix} \lambda^2 & \lambda \\ \lambda & 1 \end{pmatrix}.$$

Thus, it derives that

Theorem 3.4.1. *Let $\mathcal{L}(x,\vartheta) \in \mathbb{K}[[x]][\vartheta]^{2\times 2}$ be a non-simple matrix differential operator of second-order. Assume that a right minimal basis and a left minimal basis of its associated matrix polynomial $\mathcal{L}(0,\lambda)$ do not contain nonzero constant vectors. For any matrices $S(x)$ and $T(x)$ in $\mathbb{K}((x))^{2\times 2}$ such that $\widetilde{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ has formal power series coefficients, the matrix differential operator $\widetilde{\mathcal{L}}(x,\vartheta)$ is non-simple.*

Proof. From Lemma 3.4.1, there exist two nonsingular matrices S_0 and T_0 of $\mathbb{K}^{2\times 2}$ such that

$$S_0\,\mathcal{L}(0,\lambda)\,T_0 = \begin{pmatrix} \lambda^2 & \lambda \\ \lambda & 1 \end{pmatrix}.$$

Thus, the matrix differential operator $\mathcal{L}_1(x,\vartheta) = S_0\,\mathcal{L}(x,\vartheta)\,T_0$ satisfies the assumptions of Proposition 3.4.1. If there exist $S(x)$ and $T(x)$ in $\mathbb{K}((x))^{2\times 2}$ such that $\widetilde{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ is simple, then this implies the existence of two matrices $S_1(x) = S(x)\,S_0^{-1}$ and $T_1(x) = T_0^{-1}\,T(x)$ of $\mathbb{K}((x))^{2\times 2}$ such that $S_1(x)\,\mathcal{L}_1(x,\vartheta)\,T_1(x)$ is simple (indeed, $S_1(x)\,\mathcal{L}_1(x,\vartheta)\,T_1(x) = \widetilde{\mathcal{L}}(x,\vartheta)$) which is in contradiction with Proposition 3.4.1. □

In the next section, we provide a necessary condition on the existence of a linear substitution $y(x) = T(x)\,z(x)$ with $T(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ such that the new system $(\mathcal{L}(x,\vartheta)\,T(x))(z(x)) = 0$ is simple. An algorithm deciding the existence of such a linear substitution and computing it explicitly, when it exists, is developed. Note that in this case, the regular solutions of the original system are easily obtained by multiplying those of the simple one on the left by $T(x)$. In Section 3.6, the case where such a linear substitution does not exist is investigated. We propose a differential variant of the *EG'-method* developed in [4]: the latter algorithm can only be applied to systems with polynomial coefficients. It consists in performing elementary operations on the equations of the input system and always yields a simple linear differential system $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$ having among its regular solutions the ones of the input one. Note that the order of $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$ may be greater than ℓ and that $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$ is not necessarily equivalent to $\mathcal{L}(x,\vartheta)(y(x)) = 0$ in the sense that the regular formal solution spaces of these two systems are not necessarily isomorphic. However, we will explain at the end of Section 3.6 how one can obtain the regular solutions of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ from those of $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$.

3.5 Reduction to the simple case by linear substitutions

Theorem 3.4.1 above shows that for non-simple linear differential systems of the form (3.1) and order $\ell \geq 2$, there do not always exist two matrices $S(x)$ and $T(x)$ in $\mathrm{GL}_n(\mathbb{K}((x)))$ such that the linear differential system $(S(x)\,\mathcal{L}(x,\vartheta)\,T(x))(z(x)) = 0$ is simple. In this section, we are merely interested in the existence of a linear substitution $y(x) = T(x)\,z(x)$ with $T(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ such that the system $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$, where $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T(x)$, is simple.

3.5.1 Necessary condition for the existence of a linear substitution

We give here a necessary condition for the existence of a linear substitution $T(x)$ transforming a non-simple system into a simple one. For this, we will make use of the following lemma.

Lemma 3.5.1 ([77], Lemma 1)**.** *Every invertible matrix $T(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ can be written as*

$$T(x) = P(x)\,x^{\alpha}\,Q(x),$$

where $P(x) \in \mathbb{K}[x]^{n\times n}$ with $\det(P(x)) = 1$, $Q(x) \in \mathbb{K}[[x]]^{n\times n}$ with $\det(Q(0)) \neq 0$ and $\alpha = \mathrm{diag}(\alpha_1 I_{n_1}, \ldots, \alpha_s I_{n_s})$ where the α_i's are integers satisfying $\alpha_1 < \cdots < \alpha_s$ and the n_i's are positive integers such that $\sum_{i=1}^{s} n_i = n$.

Following [77] and using the notation of Lemma 3.5.1, we will refer to $\alpha_s - \alpha_1$ as the *span* of T and denote it by $\mathrm{span}(T)$ ($\mathrm{span}(T) = \alpha_s - \alpha_1 = -(v(T) + v(T^{-1}))$). This quantity is also called *lag* of T in [7].

Lemma 3.5.2 ([7], Prop. 1)**.** $\forall\, T_1, T_2 \in \mathrm{GL}_n(\mathbb{K}((x)))$, $\mathrm{span}(T_1\,T_2) \leq \mathrm{span}(T_1) + \mathrm{span}(T_2)$.

We give in the following theorem a necessary condition for the existence of a linear substitution leading to a simple system.

Theorem 3.5.1. *Let $\mathcal{L}(x,\vartheta)(y(x)) = 0$ be a non-simple linear differential system of the form (3.1). If there exists a linear substitution $y(x) = T(x)\,z(x)$ with $T(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ such that $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T(x) \in \mathbb{K}[[x]][\vartheta]^{n\times n}$ and the system $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ is simple, then the elements of a right minimal basis of $\mathcal{L}(0,\lambda)$ are contained in $\mathbb{K}^n$,* i.e.*, the right minimal indices of $\mathcal{L}(0,\lambda)$ are all equal to zero.*

Proof. Write $T(x) = P(x)\,x^{\alpha}\,Q(x)$ where $P(x)$, α and $Q(x)$ are as in Lemma 3.5.1. If $v(T) \geq 0$, the matrix polynomial $\overline{\mathcal{L}}(0,\lambda)$ is then equal to $\mathcal{L}(0,\lambda)\,T(0)$ and hence it is singular which is in contradiction with the hypotheses of the theorem. Therefore, there exists $k \in \{1,\ldots,s\}$ such that $\alpha_1 < \cdots < \alpha_k < 0 \leq \alpha_{k+1} < \cdots < \alpha_s$. Let $\mathcal{L}_1(x,\vartheta) = \mathcal{L}(x,\vartheta)\,P(x)$. Since $\mathcal{L}(x,\vartheta) \in \mathbb{K}[[x]][\vartheta]^{n\times n}$ and $P(x) \in \mathbb{K}[[x]]^{n\times n}$, we have $\mathcal{L}_1(x,\vartheta) \in \mathbb{K}[[x]][\vartheta]^{n\times n}$. The matrix polynomial $\mathcal{L}_1(0,\lambda)$, which is equal to $\mathcal{L}(0,\lambda)\,P(0)$, is singular (since $\mathcal{L}(0,\lambda)$ is so) and $P(0)$ is invertible, so from Lemma 1.7.1 of Chapter 1, the task reduces to proving that all the right minimal indices of $\mathcal{L}_1(0,\lambda)$ are zero. Let $\mathcal{L}_2(x,\vartheta) = \overline{\mathcal{L}}(x,\vartheta)\,Q^{-1}(x)$. Since $Q(0) \in \mathrm{GL}_n(\mathbb{K})$, the matrix inverse $Q^{-1}(x)$ belongs to $\mathbb{K}[[x]]^{n\times n}$ and hence $\mathcal{L}_2(x,\vartheta) \in \mathbb{K}[[x]][\vartheta]^{n\times n}$. The matrix polynomial $\mathcal{L}_2(0,\lambda)$ is thus equal to $\overline{\mathcal{L}}(0,\lambda)\,Q^{-1}(0)$ and consequently, it is regular. Consider now the two matrix differential operators $\mathcal{L}_2(x,\vartheta)$ and $\mathcal{L}_1(x,\vartheta)$. They are connected by $\mathcal{L}_2(x,\vartheta) = \mathcal{L}_1(x,\vartheta)\,x^{\alpha}$. Let $m = \sum_{i=1}^{k} n_i$. Since for $i = 1,\ldots,k$, $\alpha_i < 0$, the valuations in x of the first m columns of $\mathcal{L}_1(x,\vartheta)$ are necessarily positive which implies that, for $j = 1,\ldots,m$, $\mathcal{L}_1(0,\lambda)(.,j) = 0$. Consequently, the number of right minimal indices of $\mathcal{L}_1(0,\lambda)$ equal to zero is greater than or equal to m. If $m = n$ or equivalently $k = s$, then the proof ends. Otherwise, the columns $\mathcal{L}_2(0,\lambda)(.,j)$ for $j = m+1,\ldots,n$ cannot be zero since $\mathcal{L}_2(0,\lambda)$ is regular. This entails that $\alpha_{k+1} = \cdots = \alpha_s = 0$ and hence $\mathcal{L}_2(0,\lambda)(.,j) = \mathcal{L}_1(0,\lambda)(.,j)$ for $j = m+1,\ldots,n$. This proves that the dimension of the right nullspace of $\mathcal{L}_1(0,\lambda)$ is exactly equal to m since the columns $\mathcal{L}_2(0,\lambda)(.,j)$ for $j = m+1,\ldots,n$ are linearly independent. □

3.5.2 Algorithm

Let $\mathcal{L}(x,\vartheta)(y(x)) = 0$ be a non-simple linear differential system of the form (3.1) with invertible leading coefficient matrix $A_\ell(x) \in \mathbb{K}[x]^{n\times n}$. We suppose that we have already simplified the equations of the system by suitable powers of x so that we can have $v(\mathcal{L}(x,\vartheta)(i,.)) = 0$ for $i = 1,\ldots,n$. In other terms, we suppose that all the rows of $\mathcal{L}(0,\lambda)$ are nonzero. We will now develop an algorithm that either computes a linear substitution $y(x) = T(x)\,z(x)$ such that the new system $(\mathcal{L}(x,\vartheta)\,T(x))(z(x)) = 0$ is simple or proves that such a linear substitution does not exist. It proceeds as follows. First, compute a right minimal basis of $\mathcal{L}(0,\lambda)$. If one of its elements is non constant, *i.e.*, belongs to $\mathbb{K}[\lambda]^n \setminus \mathbb{K}^n$, then, by Theorem 3.5.1, such a linear substitution does not exist and we are done. Otherwise, let $\mathcal{B}$ denote the matrix whose columns are the elements of the computed right minimal basis. For every column $\mathcal{B}(.,k)$, select one of its nonzero entries, say $\mathcal{B}(i_k,k)$, in such a way that the degree of the i_kth column of $A_\ell(x)$ is maximal among the degrees of the columns of $A_\ell(x)$ of indices corresponding to the nonzero entries of $\mathcal{B}(.,k)$. Then, execute the following *reduction procedure*:

1. Replace the i_kth column of $\mathcal{L}(x,\vartheta)$ by $\mathcal{L}(x,\vartheta)\,\mathcal{B}(.,k)$. This is equivalent to multiplying $\mathcal{L}(x,\vartheta)$ on the right by an invertible constant matrix T_1 obtained from the identity matrix I_n by replacing its i_kth column by the column $\mathcal{B}(.,k)$. If $\widetilde{\mathcal{L}}(x,\vartheta)$ denotes the resulting operator, then its associated matrix polynomial $\widetilde{\mathcal{L}}(0,\lambda)$ is equal to $\mathcal{L}(0,\lambda)\,T_1$ and hence the i_kth column of $\widetilde{\mathcal{L}}(0,\lambda)$ is zero. Note here that all rows of $\widetilde{\mathcal{L}}(0,\lambda)$ are nonzero since T_1 is invertible and the rows of $\mathcal{L}(0,\lambda)$ are supposed to be nonzero as well. As for the i_kth

column of the leading coefficient matrix $\widetilde{A}_\ell(x)$ of $\widetilde{\mathcal{L}}(x,\vartheta)$, it is exactly equal to $A_\ell(x)\,\mathcal{B}(.,k)$ since $\widetilde{A}_\ell(x) = A_\ell(x)\,T_1$;

2. Let $\gamma_{i_k} = v\Big(\widetilde{\mathcal{L}}(x,\vartheta)(.,i_k)\Big)$ be the valuation of the i_kth column of $\widetilde{\mathcal{L}}(x,\vartheta)$. Since the leading coefficient $A_\ell(x)$ of $\mathcal{L}(x,\vartheta)$ is assumed to be invertible, this guarantees that $\widetilde{\mathcal{L}}(x,\vartheta)(.,i_k)$ is a nonzero column and implies that γ_{i_k} is finite, positive and less than or equal to the degree of the i_kth column of $\widetilde{A}_\ell(x)$. Multiply now each component of the i_kth column of $\widetilde{\mathcal{L}}(x,\vartheta)$ on the right by $x^{-\gamma_{i_k}}$ (note that $\vartheta^j x^{-\gamma_{i_k}} = x^{-\gamma_{i_k}}(\vartheta-\gamma_{i_k})^j$). This is equivalent to multiplying $\widetilde{\mathcal{L}}(x,\vartheta)$ on the right by an invertible diagonal matrix $T_2 \in \mathbb{K}[x^{-1}]^{n\times n}$ obtained from the identity matrix I_n by replacing its i_kth diagonal entry by $x^{-\gamma_{i_k}}$. Let $\overline{\mathcal{L}}(x,\vartheta) = \widetilde{\mathcal{L}}(x,\vartheta)\,T_2$. By definition of γ_{i_k}, the i_kth column of $\overline{\mathcal{L}}(0,\lambda)$ is nonzero. Here again, the rows of $\overline{\mathcal{L}}(0,\lambda)$ are nonzero which means that a simplification of the rows of $\overline{\mathcal{L}}(x,\vartheta)$ by powers of x cannot occur.

Now, we use $\mathcal{B}(i_k,k)$ as a pivot to eliminate all the elements $\mathcal{B}(i_k,j)$ for $j \neq k$. In this way, the new columns of $\mathcal{B}$ of index $j \neq k$ belong now to the right nullspace of $\overline{\mathcal{L}}(0,\lambda)$. We then repeat the reduction procedure on $\overline{\mathcal{L}}(x,\vartheta)$ using the new columns of $\mathcal{B}$ and so on.

Proposition 3.5.1. *The reduction procedure described above strictly reduces the degree of one column of the leading coefficient matrix while the degrees of the other columns remain unchanged.*

Proof. The i_kth column of $\overline{\mathcal{L}}(x,\vartheta)$ is given by

$$\overline{\mathcal{L}}(x,\vartheta)(.,i_k) = \mathcal{L}(x,\vartheta)\,\mathcal{B}(.,k)\,x^{-\gamma_{i_k}} = x^{-\gamma_{i_k}} \sum_{i=0}^{\ell} A_i(x)\,\mathcal{B}(.,k)\,(\vartheta-\gamma_{i_k})^i. \tag{3.16}$$

From the relation (3.16), we can deduce that the degree of the i_kth column of the leading coefficient $\overline{A}_\ell(x)$ of $\overline{\mathcal{L}}(x,\vartheta)$ is less than or equal to that of the i_kth column of $A_\ell(x)$ minus γ_{i_k}. Indeed, let d_j, for $j = 1,\ldots,n$, denote the degree of the jth column of $A_\ell(x)$ and write $A_\ell(x) = \sum_{i=0}^{d} A_{\ell,i}\,x^i$ where $d = \max_{j=1,\ldots,n} d_j$. According to (3.16), the i_kth column of $\overline{A}_\ell(x)$ is defined as follows:

$$\overline{A}_\ell(x)(.,i_k) = A_\ell(x)\,\mathcal{B}(.,k)\,x^{-\gamma_{i_k}} = \sum_{i=0}^{d} A_{\ell,i}\,\mathcal{B}(.,k)\,x^{i-\gamma_{i_k}}.$$

By definition of $\gamma_{i_k} > 0$, we have $A_{\ell,i}\,\mathcal{B}(.,k) = 0$, for $i = 0,\ldots,\gamma_{i_k}-1$. Moreover, for $j \in \{1,\ldots,n\}$ such that $d_j > d_{i_k}$, we have $\mathcal{B}(j,k) = 0$ (from the choice of i_k), and for $j \in \{1,\ldots,n\}$ such that $d_j \leq d_{i_k}$ and for i such that $d_{i_k} < i \leq d$, we have $A_{\ell,i}(.,j) = 0$. Consequently, for $d_{i_k} < i \leq d$, $A_{\ell,i}\,\mathcal{B}(.,k) = \sum_{j=1}^{n} A_{\ell,i}(.,j)\,\mathcal{B}(j,k) = 0$ and then $\deg(\overline{A}_\ell(x)(.,i_k)) \leq d_{i_k} - \gamma_{i_k} < d_{i_k} = \deg(A_\ell(x)(.,i_k))$. □

We illustrate the above approach with the following example:

Example 3.5.1. *Consider the matrix differential operator given by*

$$\mathcal{L}(x,\vartheta) = A_2(x)\,\vartheta^2 + A_1(x)\,\vartheta + A_0(x) = \begin{pmatrix} 1+x^2 & 2 & 1 \\ 0 & 3\,x & 4\,x \\ 0 & 0 & x \end{pmatrix}\vartheta^2 + \begin{pmatrix} x & x^2 & 0 \\ 1 & 2 & 1 \\ 0 & x^2 & 0 \end{pmatrix}\vartheta + \begin{pmatrix} 1+x & 2 & 1 \\ 0 & x^2 & 0 \\ 2+x & 4 & 2 \end{pmatrix}.$$

This operator is non-simple since its associated matrix polynomial

$$\mathcal{L}(0,\lambda) = \begin{pmatrix} \lambda^2+1 & 2\,\lambda^2+2 & \lambda^2+1 \\ \lambda & 2\,\lambda & \lambda \\ 2 & 4 & 2 \end{pmatrix}$$

is singular. A right minimal basis of $\mathcal{L}(0,\lambda)$ *is given by the columns of the matrix*

$$\mathcal{B} = \begin{pmatrix} -2 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

We consider the first column of $\mathcal{B}$ *which corresponds to* $k = 1$ *with the previous notation. Its first two components are nonzero, but since* $\deg(A_2(x)(.,1)) = 2 > \deg(A_2(x)(.,2)) = 1$*, we select the first one,* i.e., $i_1 = 1$*. We then apply our reduction procedure. We first replace* $\mathcal{L}(x,\vartheta)(.,1)$ *by* $\mathcal{L}(x,\vartheta)\,\mathcal{B}(.,1)$*,* i.e., *multiply operator* $\mathcal{L}(x,\vartheta)$ *on the right by*

$$T_1 = \begin{pmatrix} -2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The operator obtained is given by

$$\mathcal{L}_1(x,\vartheta) = \begin{pmatrix} -2\,x^2 & 2 & 1 \\ 3\,x & 3\,x & 4\,x \\ 0 & 0 & x \end{pmatrix}\vartheta^2 + \begin{pmatrix} -2\,x + x^2 & x^2 & 0 \\ 0 & 2 & 1 \\ x^2 & x^2 & 0 \end{pmatrix}\vartheta + \begin{pmatrix} -2\,x & 2 & 1 \\ x^2 & x^2 & 0 \\ -2\,x & 4 & 2 \end{pmatrix}.$$

The first column of $\mathcal{L}_1(0,\lambda)$ *is now zero and the degree of the first column of the leading coefficient matrix has not changed. With the previous notation, we have* $\gamma_1 = 1$ *so we multiply* $\mathcal{L}_1(x,\vartheta)(.,1)$ *on the right by* x^{-1} *which corresponds to multiplying* $\mathcal{L}_1(x,\vartheta)$ *on the right by the matrix* $T_2(x) = \mathrm{diag}(x^{-1},1,1)$*. We obtain the new matrix differential operator*

$$\mathcal{L}_2(x,\vartheta) = \begin{pmatrix} -2\,x & 2 & 1 \\ 3 & 3\,x & 4\,x \\ 0 & 0 & x \end{pmatrix}\vartheta^2 + \begin{pmatrix} 5\,x - 2 & x^2 & 0 \\ -6 & 2 & 1 \\ x & x^2 & 0 \end{pmatrix}\vartheta + \begin{pmatrix} -3\,x & 2 & 1 \\ 3 + x & x^2 & 0 \\ -2 - x & 4 & 2 \end{pmatrix}.$$

Consequently, the first column of $\mathcal{L}_2(0,\lambda)$ *is nonzero and the degree of the first column of the leading coefficient matrix has decreased by* $\gamma_1 = 1$*. Then, we use* $\mathcal{B}(1,1)$ *as a pivot to eliminate* $\mathcal{B}(1,2)$ *and obtain*

$$\mathcal{B}_1 = \begin{pmatrix} -2 & 0 \\ 1 & -\frac{1}{2} \\ 0 & 1 \end{pmatrix}.$$

We now consider the second column of $\mathcal{B}_1$ *which belongs to a right minimal basis of* $\mathcal{L}_2(0,\lambda)$*. Since the degrees in* x *of the second and third columns of the leading coefficient of* $\mathcal{L}_2(x,\vartheta)$ *are equal, we choose one of these two columns to perform our reduction. Let us choose the second one. Replacing* $\mathcal{L}_2(x,\vartheta)(.,2)$ *by* $\mathcal{L}_2(x,\vartheta)\,\mathcal{B}_1(.,2)$*, we get*

$$\mathcal{L}_3(x,\vartheta) = \begin{pmatrix} -2\,x & 0 & 1 \\ 3 & \frac{5}{2}x & 4\,x \\ 0 & x & x \end{pmatrix}\vartheta^2 + \begin{pmatrix} 5\,x - 2 & -\frac{1}{2}\,x^2 & 0 \\ -6 & 0 & 1 \\ x & -\frac{1}{2}\,x^2 & 0 \end{pmatrix}\vartheta + \begin{pmatrix} -3\,x & 0 & 1 \\ 3 + x & -\frac{1}{2}\,x^2 & 0 \\ -2 - x & 0 & 2 \end{pmatrix}.$$

Now we have $v(\mathcal{L}_3(x,\vartheta)(.,2)) = 1$*. So we multiply* $\mathcal{L}_3(x,\vartheta)(.,2)$ *on the right by* x^{-1}*. We obtain the matrix differential operator*

$$\mathcal{L}_4(x,\vartheta) = \begin{pmatrix} -2\,x & 0 & 1 \\ 3 & \frac{5}{2} & 4\,x \\ 0 & 1 & x \end{pmatrix}\vartheta^2 + \begin{pmatrix} 5\,x - 2 & -\frac{1}{2}\,x & 0 \\ -6 & -5 & 1 \\ x & -2 - \frac{1}{2}\,x & 0 \end{pmatrix}\vartheta + \begin{pmatrix} -3\,x & \frac{1}{2}\,x & 1 \\ 3 + x & \frac{5}{2} - \frac{1}{2}\,x & 0 \\ -2 - x & 1 + \frac{1}{2}\,x & 2 \end{pmatrix}$$

which is connected to $\mathcal{L}_2(x,\vartheta)$ *by* $\mathcal{L}_4(x,\vartheta) = \mathcal{L}_2(x,\vartheta)\,T_3(x)$ *with*

$$T_3(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{2\,x} & 0 \\ 0 & \frac{1}{x} & 1 \end{pmatrix}.$$

The determinant of the matrix polynomial $\mathcal{L}_4(0,\lambda)$ *is equal to* $(\lambda-1)^2\,(3\,\lambda^4-6\,\lambda^3+13\,\lambda^2-16\,\lambda+8)$ *and hence operator* $\mathcal{L}_4(x,\vartheta)$ *is simple. We finally note that the two operators* $\mathcal{L}(x,\vartheta)$ *and* $\mathcal{L}_4(x,\vartheta)$ *are related by* $\mathcal{L}_4(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T(x)$ *with*

$$T(x) = T_1\,T_2(x)\,T_3(x) = \begin{pmatrix} -\frac{2}{x} & 0 & 0 \\ \frac{1}{x} & -\frac{1}{2\,x} & 0 \\ 0 & \frac{1}{x} & 1 \end{pmatrix}.$$

■

The following corollary follows from Proposition 3.5.1.

Corollary 3.5.1. *The number of iterations of the reduction procedure described above does not exceed* $D = \sum_{j=1}^{n} \deg(A_\ell(x)(.,j))$.

Thus, after applying the reduction procedure at most D times, we obtain a matrix differential operator $\widehat{\mathcal{L}}(x,\vartheta)$ which can be written as $\mathcal{L}(x,\vartheta)\,T(x)$ with $T(x)$ an invertible matrix such that either $\widehat{\mathcal{L}}(x,\vartheta)$ is simple or a right minimal basis of $\widehat{\mathcal{L}}(0,\lambda)$ contains non constant elements.

Lemma 3.5.3. *The matrix* T *constructed by applying iteratively the reduction procedure described above is a matrix polynomial in* x^{-1} *satisfying* $-D \leq v(T) < 0$ *and* $\operatorname{span}(T) \leq D$ *where* $D = \sum_{j=1}^{n} \deg(A_\ell(x)(.,j))$.

Proof. The matrix T is the product of invertible matrices which are either constant matrices or diagonal matrices of the form $\operatorname{diag}(1,\ldots,1,x^{-\gamma_{i_k}},1,\ldots,1)$ where $1 \leq \gamma_{i_k} \leq \deg(A_\ell(x)(.,i_k))$. Therefore, $T \in \mathbb{K}[x^{-1}]^{n\times n}$ with $-\sum \gamma_{i_k} \leq v(T) < 0$ and $\operatorname{span}(T) \leq \sum \gamma_{i_k}$ (see Lemma 3.5.2). Now, from Proposition 3.5.1 and Corollary 3.5.1, $\sum \gamma_{i_k}$ cannot exceed D which ends the proof. □

In practice, we deal with matrix differential operators with truncated coefficient matrices. Let $\mathcal{L}(x,\vartheta)_{|N}$ denote the operator $\mathcal{L}(x,\vartheta)$ given by (3.1) truncated at order N, *i.e.*, $\mathcal{L}(x,\vartheta)_{|N} = \sum_{i=1}^{N} x^i\,L_i(\vartheta) + \mathcal{L}(0,\vartheta)$, where the L_i's are given by (3.4). The question that arises here is how to choose N so that if we apply iteratively the reduction procedure described above to $\mathcal{L}_{|N}(x,\vartheta)$ and get a matrix T such that $\mathcal{L}(x,\vartheta)_{|N}\,T$ is simple, then we can ensure that the whole operator $\mathcal{L}(x,\vartheta)\,T$ is simple too.

Proposition 3.5.2. *With the notation above, we have*

$$\forall\, N \in \mathbb{N},\ \mathcal{L}(x,\vartheta)\,T = \mathcal{L}(x,\vartheta)_{|N}\,T \mod x^{N+v(T)+1}.$$

Therefore, if $\mathcal{L}(x,\vartheta)_{|N}\,T$ *is simple with* $N \geq -v(T)$, *then so is* $\mathcal{L}(x,\vartheta)\,T$.

Proof. The first assertion is obvious. Now, for $N \geq -v(T)$, we have, in particular, $\mathcal{L}(x,\vartheta)\,T = \mathcal{L}(x,\vartheta)_{|N}\,T \mod x$. Therefore, the matrix polynomials associated respectively with $\mathcal{L}(x,\vartheta)\,T$ and $\mathcal{L}(x,\vartheta)_{|N}\,T$ are equal. Hence, $\mathcal{L}(x,\vartheta)_{|N}\,T$ is simple implies that $\mathcal{L}(x,\vartheta)\,T$ is simple too. □

From the discussion above, we derive the following algorithm:

ALGORITHM **LinSubs**

INPUT: The operator $\mathcal{L}(x,\vartheta)_{|N}$, the truncation of $\mathcal{L}(x,\vartheta)$ at order N, where N is an integer greater than or equal to $\sum_{j=1}^{n}\deg(A_\ell(x)(.,j))$.

OUTPUT: The empty list $[\,]$ in case $\mathcal{L}(x,\vartheta)$ cannot be reduced to a simple operator by means of a linear substitution or an invertible matrix $T\in\mathbb{K}[x^{-1}]^{n\times n}$ such that $\mathcal{L}(x,\vartheta)\,T$ is simple.

INITIALIZATION: $T=I_n$ and $\overline{\mathcal{L}}(x,\vartheta)=\mathcal{L}(x,\vartheta)_{|N}$;

While $\overline{\mathcal{L}}(0,\lambda)$ is singular **do**
1. Compute a matrix $\mathcal{B}$ whose columns form a right minimal basis of $\overline{\mathcal{L}}(0,\lambda)$;
2. **If** $\deg(\mathcal{B})>0$ **then**
   Return $[\,]$;
3. **else**
   For each column $\mathcal{B}(.,k)$ **do**
   3.1. Let $\overline{A}_\ell$ be the leading coefficient matrix of $\overline{\mathcal{L}}(x,\vartheta)$;
   3.2. Let $J_k=\{i\in\{1,\ldots,n\}$ such that $\mathcal{B}(i,k)\neq 0\}$;
   3.3. Choose $i_k\in J_k$ such that $\deg(\overline{A}_\ell(.,i_k))\geq\deg(\overline{A}_\ell(.,j))\ \forall\, j\in J_k$;
   3.4. Apply the following reduction procedure:
   3.4.(a). Let $\overline{\mathcal{L}}(x,\vartheta)(.,i_k)=\overline{\mathcal{L}}(x,\vartheta)\,\mathcal{B}(.,k)$;
   3.4.(b). Let $\gamma_{i_k}=v\big(\overline{\mathcal{L}}(x,\vartheta)(.,i_k)\big)$ and $\overline{\mathcal{L}}(x,\vartheta)(j,i_k)=\overline{\mathcal{L}}(x,\vartheta)(j,i_k)\,x^{-\gamma_{i_k}}$ for $j=1,\ldots,n$;
   3.5. Let $T(.,i_k)=x^{-\gamma_{i_k}}\,T\,\mathcal{B}(.,k)$;
   3.6. Use $\mathcal{B}(i_k,k)$ as a pivot to eliminate all the elements $\mathcal{B}(i_k,j)$ with $j\neq k$;
   end do;
end if;
end do;
Return T;

Proposition 3.5.3. *Let $\mathcal{L}(x,\vartheta)$ be a non-simple matrix differential operator of the form (3.1) with invertible leading coefficient matrix $A_\ell(x)\in\mathbb{K}[x]^{n\times n}$. Let $D=\sum_{j=1}^{n}\deg(A_\ell(x)(.,j))$ and $N\geq D$ be the order of truncation of the coefficient matrices $A_i(x)$ of $\mathcal{L}(x,\vartheta)$. Then, Algorithm* **LinSubs** *stops after at most D calls of the reduction procedure and uses at most $O^{\sim}\big(n^{\omega+1}\,\ell\,D+n^2\,\ell\,N\,D\big)$ arithmetic operations in $\mathbb{K}$.*

Proof. The first assertion follows from Corollary 3.5.1. Let us now study the complexity of the algorithm. From Lemma 1.7.2 of Chapter 1, computing a right minimal basis of an $n\times n$ matrix polynomial with entries degrees bounded by ℓ can be done in $O^{\sim}\big(n^{\omega+1}\,\ell\big)$ arithmetic operations in $\mathbb{K}$. In the algorithm, we compute at most D right minimal bases, so the total cost of minimal bases computations is bounded by $O^{\sim}\big(n^{\omega+1}\,\ell\,D\big)$ operations in $\mathbb{K}$. In Step 3.4.(a), write $\overline{\mathcal{L}}(x,\vartheta)=\sum_{j=0}^{\overline{N}}x^j\,\overline{L}_j(\vartheta)$ where $\overline{N}\leq N$. The cost of computing the product $\overline{\mathcal{L}}(x,\vartheta)\,\mathcal{B}(.,k)$ is thus bounded by $(\overline{N}+1)$ times the cost of computing one product $\overline{L}_j(\vartheta)\,\mathcal{B}(.,k)$. Since $\overline{L}_j(\vartheta)$ is a constant matrix operator of order at most ℓ, computing $\overline{L}_j(\vartheta)\,\mathcal{B}(.,k)$ can be done using at most $O\big(n^2\,\ell\big)$ operations in $\mathbb{K}$. Consequently, Step 3.4.(a) can be done in at most $O\big(n^2\,\ell\,N\big)$ operations in $\mathbb{K}$. In Step 3.4.(b), multiplying each component of the column $\overline{\mathcal{L}}(x,\vartheta)(.,i_k)$ on the right by $x^{-\gamma_{i_k}}$ can be done by substituting ϑ by $\vartheta-\gamma_{i_k}$ in the i_kth column of each matrix $\overline{L}_j(\vartheta)$. The latter operation uses at most $O(n\,\ell)$ operations in $\mathbb{K}$. Since we have at most $N+1$ matrices

$\overline{L}_j(\vartheta)$, then the total cost of Step 3.4.(b) is at most $O(n\,\ell\,N)$ operations in $\mathbb{K}$. Consequently, the reduction procedure can be done using at most $O\big(n^2\,\ell\,N\big)$ operations in $\mathbb{K}$. Concerning Step 3.5, since $T \in \mathbb{K}[x^{-1}]^{n\times n}$ of degree in x^{-1} always bounded by D, Step 3.5 can then be performed using at most $O\big(n^2\,D\big)$ operations in $\mathbb{K}$. Finally Step 3.6 uses at most $O\big(n^2\big)$ operations in $\mathbb{K}$. Thus, one passage by the **For** loop costs at most $O\big(n^2\,\ell\,N\big)$ operations in $\mathbb{K}$. Consequently, the total cost of the algorithm is at most $O^{\sim}\big(n^{\omega+1}\,\ell\,D + n^2\,\ell\,N\,D\big)$ arithmetic operations in $\mathbb{K}$. □

3.5.3 Reconstruction of the regular solutions

The following proposition shows how to choose N in Algorithm **LinSubs** and ν in Algorithm **BCE_V2** in order to get the general regular solution of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ up to a fixed order ν_1.

Proposition 3.5.4. *Let $\mathcal{L}(x,\vartheta)(y(x)) = 0$ be a non-simple linear differential system of the form (3.1) having an invertible leading coefficient matrix $A_\ell(x) \in \mathbb{K}[x]^{n\times n}$. Suppose that $\mathcal{L}(x,\vartheta)$ can be reduced to a simple operator by means of a linear substitution. Let $D = \sum_{j=1}^{n} \deg(A_\ell(x)(.,j))$ and $\nu_1 \in \mathbb{N}$. The general regular solution up to order ν_1 of system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ can be computed as follows:*

1. *apply Algorithm* **LinSubs** *to $\mathcal{L}(x,\vartheta)_{|N}$ with $N = \nu_1 + 2D$. Let T be the computed matrix; then*
2. *apply Algorithm* **BCE_V2** *to $\mathcal{L}(x,\vartheta)_{|N}\,T$ with $\nu = \nu_1 + D$, and finally,*
3. *multiply the output of Algorithm* **BCE_V2** *on the left by T.*

Proof. We will start by proving that to compute the regular solutions of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ up to order ν_1, it suffices to compute those of $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$, where $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T$, up to order $\nu_1 + D$. Write $T = P(x)\,x^{\alpha}\,Q(x)$ where $P(x) \in \mathbb{K}[x]^{n\times n}$ and $Q(x) \in \mathbb{K}[[x]]^{n\times n}$ are as in Lemma 3.5.1 and $\alpha = \mathrm{diag}(\alpha_1,\ldots,\alpha_n)$ with $\alpha_1 \le \cdots \le \alpha_n$ and $\alpha_1 < 0$ because $v(T) < 0$. Put $\mathcal{L}_1(x,\vartheta) = \mathcal{L}(x,\vartheta)\,P(x)$ and $\mathcal{L}_2(x,\vartheta) = \overline{\mathcal{L}}(x,\vartheta)\,Q^{-1}(x)$, then $\mathcal{L}_2(x,\vartheta) = \mathcal{L}_1(x,\vartheta)\,x^{\alpha}$. Since $P(x)$ is unimodular ($P(0)$ is invertible), then to each regular solution y of $\mathcal{L}(x,\vartheta)$ of exponent $\lambda_0 \in \overline{\mathbb{K}}$ corresponds a regular solution $u = P^{-1}(x)\,y$ of $\mathcal{L}_1(x,\vartheta)$ of the same exponent λ_0. Moreover, to have y up to order ν_1, it suffices to compute u up to order ν_1. Similar statements hold for the regular solutions z of $\overline{\mathcal{L}}(x,\vartheta)$ and those $w = Q(x)\,z$ of $\mathcal{L}_2(x,\vartheta)$. Now, u and w are related by $u = x^{\alpha}\,w$. Our problem is now reduced to show that to compute u up to order ν_1 it suffices to have w up to order $\nu_1 + D$. For this, write $w = x^{\lambda_1}\widetilde{w}$ with $\lambda_1 \in \overline{\mathbb{K}}$ and

$$\widetilde{w} = \widetilde{W}_0 + \widetilde{W}_1\,x + \cdots + \widetilde{W}_k\,x^k + \cdots, \quad \text{where } \widetilde{W}_k \in \overline{\mathbb{K}}[\log(x)]^n \text{ and } \widetilde{W}_0 \neq 0.$$

Let $\widetilde{w}_i$ denote the ith component of $\widetilde{w}$ and $v_i = v(\widetilde{w}_i) \ge 0$. Put $\widetilde{u} = x^{\alpha}\,\widetilde{w}$ and $m = v(\widetilde{u}) = \min_{i=1,\ldots,n}(\alpha_i + v_i)$. Since $\widetilde{W}_0 \neq 0$, then there exists $i_0 \in \{1,\ldots,n\}$ such that the i_0th component of $\widetilde{W}_0$ is nonzero, *i.e.*, for which $v_{i_0} = 0$. Therefore, m is well defined and $\alpha_1 \le m \le \alpha_{i_0} + v_{i_0} = \alpha_{i_0} \le \alpha_n$. Since $u = x^{\alpha}\,w = x^{\lambda_1}\,x^{\alpha}\,\widetilde{w} = x^{\lambda_1}\,\widetilde{u}$, u is then a regular solution of $\mathcal{L}_1(x,\vartheta)$ of exponent $\lambda_1 + m$. Write

$$\widetilde{u} = x^m\left(\widetilde{U}_0 + \widetilde{U}_1\,x + \cdots + \widetilde{U}_k\,x^k + \cdots\right), \quad \text{where } \widetilde{U}_k \in \overline{\mathbb{K}}[\log(x)]^n \text{ and } \widetilde{U}_0 \neq 0.$$

Computing u up to order ν_1 requires the knowledge of the coefficients $\widetilde{U}_0,\ldots,\widetilde{U}_{\nu_1}$. Now, the ith component $\left(\widetilde{U}_k\right)_i$ of $\widetilde{U}_k$ is the coefficient of x^{k+m} in $\widetilde{u}_i = x^{\alpha_i}\,\widetilde{w}_i$, hence, $\left(\widetilde{U}_k\right)_i = \left(\widetilde{W}_{k+m-\alpha_i}\right)_i$. Thus, the coefficients $\widetilde{U}_0,\ldots,\widetilde{U}_{\nu_1}$ depend on $\widetilde{W}_0,\ldots,\widetilde{W}_{\nu_1+m-\alpha_1}$. Hence, it suffices to compute w up to order $\nu_1 + D$ because we have $\nu_1 + m - \alpha_1 \le \nu_1 + \alpha_n - \alpha_1 = \nu_1 + \mathrm{span}(T) \le \nu_1 + D$

(see Lemma 3.5.3). Now, since $\overline{\mathcal{L}}(x,\vartheta)$ is simple, we only have to consider $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)T$ truncated at order ν_1+D (see Remark 2.6.1 of Chapter 2). Thus, according to Proposition 3.5.2, we need to truncate $\mathcal{L}(x,\vartheta)$ at order $\nu_1+D-v(T)$, so we take $N=\nu_1+2D \geq \nu_1+D-v(T)$. □

3.6 A differential variant of the EG'-algorithm

In this section, we consider a non-simple linear differential system of the form (3.1) with invertible leading coefficient $A_\ell(x)$ and we suppose that all its coefficient matrices $A_i(x)$ **are $n\times n$ matrix polynomials**. Inspired by the *EG'-algorithm* proposed by Abramov et al. in [4] (see also [1, 2, 3]), we develop an algorithm which carries out elementary operations on the rows of $\mathcal{L}(x,\vartheta)$ and always yields a simple operator $\overline{\mathcal{L}}(x,\vartheta)$ which can be written as $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{P}(x,\vartheta)\mathcal{L}(x,\vartheta)$ with $\mathcal{P}(x,\vartheta) \in \mathbb{K}[x^{-1}][\vartheta]^{n\times n}$. We then explain how to recover the regular solutions of $\mathcal{L}(x,\vartheta)$ from those of $\overline{\mathcal{L}}(x,\vartheta)$ which can be computed using Algorithm **BCE_V2**.

3.6.1 Definitions and preliminaries

In the sequel, we use definitions and terminologies defined in [27] for matrices of Ore polynomials and we adapt them to matrix differential operators in $\mathbb{K}((x))[\vartheta]^{n\times n}$.

Definition 3.6.1. *Let $\mathcal{L}(x,\vartheta) \in \mathbb{K}((x))[\vartheta]^{n\times n}$ be a matrix differential operator and $J \subseteq \{1,\ldots,n\}$. The rows of $\mathcal{L}(x,\vartheta)$ with index $i \in J$ are said to be* $\mathbb{K}((x))[\vartheta]$-linearly dependent *if there exist differential operators $\{W_i\}_{i\in J}$ in $\mathbb{K}((x))[\vartheta]$ not all zero such that $\sum_{i\in J} W_i\,\mathcal{L}(x,\vartheta)(i,.) = 0$. Otherwise, they are said to be* $\mathbb{K}((x))[\vartheta]$-linearly independent.

Definition 3.6.2. *Let $\mathcal{L}(x,\vartheta) \in \mathbb{K}((x))[\vartheta]^{n\times n}$ be a matrix differential operator. Denote by $\mathcal{M}_{\mathcal{L}}$ the submodule of the left $\mathbb{K}((x))[\vartheta]$-module $\mathbb{K}((x))[\vartheta]^{1\times n}$ defined by*

$$\mathcal{M}_{\mathcal{L}} = \left\{P(x,\vartheta)\,\mathcal{L}(x,\vartheta);\ P(x,\vartheta) \in \mathbb{K}((x))[\vartheta]^{1\times n}\right\}.$$

The row rank *of $\mathcal{L}(x,\vartheta)$ is defined to be the rank of the module $\mathcal{M}_{\mathcal{L}}$ which is, following [38, page 28], equal to the cardinality of a maximal $\mathbb{K}((x))[\vartheta]$-linearly independent subset of $\mathcal{M}_{\mathcal{L}}$.*

It has been shown in [27, Appendix] that the row rank of $\mathcal{L}(x,\vartheta)$ is equal to the maximum number of $\mathbb{K}((x))[\vartheta]$-linearly independent rows of $\mathcal{L}(x,\vartheta)$. In the rest of this section, we will refer to the row rank by simply *rank*.

In this section, we are merely interested in applying two types of elementary row operations to a matrix differential operator $\mathcal{L}(x,\vartheta)$. The first type of elementary row operations includes:

(E1) interchanging two rows of $\mathcal{L}(x,\vartheta)$;

(E2) adding to a row of $\mathcal{L}(x,\vartheta)$ another row multiplied on the left by a scalar differential operator in $\mathbb{K}((x))[\vartheta]$;

(E3) multiplying a row of $\mathcal{L}(x,\vartheta)$ on the left by a nonzero scalar differential operator in $\mathbb{K}((x))[\vartheta]$;

The second type includes the elementary row operations (E1) and (E2) and

(E3') multiplying a row of $\mathcal{L}(x,\vartheta)$ on the left by a nonzero element of $\mathbb{K}((x))$.

Note that each elementary row operation can be performed by multiplying $\mathcal{L}(x,\vartheta)$ on the left by a square matrix differential operator.

Definition 3.6.3. *1. A matrix differential operator* $\mathcal{P}(x,\vartheta) \in \mathbb{K}((x))[\vartheta]^{n\times n}$ *is said to be* unimodular *if it has a two-sided inverse in* $\mathbb{K}((x))[\vartheta]^{n\times n}$*, that is, if there exists a matrix differential operator* $\mathcal{Q}(x,\vartheta) \in \mathbb{K}((x))[\vartheta]^{n\times n}$ *such that* $\mathcal{Q}(x,\vartheta)\,\mathcal{P}(x,\vartheta) = \mathcal{P}(x,\vartheta)\,\mathcal{Q}(x,\vartheta) = I_n$*.*

2. Two matrix differential operators $\mathcal{L}(x,\vartheta)$ *and* $\overline{\mathcal{L}}(x,\vartheta)$ *are said to be* left-equivalent *if there exists a unimodular matrix differential operator* $\mathcal{P}(x,\vartheta)$ *such that* $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{P}(x,\vartheta)\,\mathcal{L}(x,\vartheta)$*.*

It is easy to see that two left-equivalent matrix differential operators have the same regular formal solution space.

Lemma 3.6.1 ([75], Theorem III)**.** *A matrix differential operator* $\mathcal{P}(x,\vartheta)$ *is unimodular if and only if it can be expressed as a product of elementary operations of the second type.*

Thus, two matrix differential operators $\mathcal{L}(x,\vartheta)$ and $\overline{\mathcal{L}}(x,\vartheta)$ are left-equivalent if one is obtained from the other by means of elementary row operations of the second type.

Lemma 3.6.2. *[27, Lemma A.3] The rank of a matrix differential operator* $\mathcal{L}(x,\vartheta)$ *does not change if one applies to* $\mathcal{L}(x,\vartheta)$ *elementary row operations of the first or of the second type.*

Proposition 3.6.1. *A matrix differential operator* $\mathcal{L}(x,\vartheta) = \sum_{i=0}^{\ell} A_i(x)\vartheta^i \in \mathbb{K}((x))[\vartheta]^{n\times n}$ *with* $A_\ell(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ *is of rank* n*.*

Proof. Since $A_\ell(x)$ is an invertible matrix, we may suppose, without loss of generality, that it is the identity matrix I_n. If the rows of $\mathcal{L}(x,\vartheta)$ are $\mathbb{K}((x))[\vartheta]$-linearly dependent, then there exist $W_1,\ldots,W_n \in \mathbb{K}((x))[\vartheta]$ not all zero such that $\sum_{i=1}^{n} W_i\,\mathcal{L}(x,\vartheta)(i,.) = 0$. Since the leading coefficient of $\mathcal{L}(x,\vartheta)$ is supposed to be the identity matrix, the order of each diagonal entry $\mathcal{L}(x,\vartheta)(i,i)$ is greater than those of other entries $\mathcal{L}(x,\vartheta)(i,j)$ for $j \neq i$. Choose $j_0 \in \{1,\ldots,n\}$ such that the order of the differential operator W_{j_0} is greater than or equal to the orders of all the W_j for $j \neq j_0$. We have $W_{j_0}\,\mathcal{L}(x,\vartheta)(j_0,j_0) = -\sum_{i\neq j_0} W_i\,\mathcal{L}(x,\vartheta)(i,j_0)$ which is impossible since the order of the left-hand side of the latter equality is greater than the order of its right-hand side. □

3.6.2 Algorithm

The following algorithm consists in applying elementary row operations of the first or second type to a non-simple matrix differential operator $\mathcal{L}(x,\vartheta) \in \mathbb{K}[x][\vartheta]^{n\times n}$ with invertible leading coefficient and yields another operator $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{P}(x,\vartheta)\,\mathcal{L}(x,\vartheta)$ whose rank is equal to that of $\overline{\mathcal{L}}(0,\lambda)$. We know, from Proposition 3.6.1, that the rank of $\mathcal{L}(x,\vartheta)$ is equal to n. Consequently, from Lemma 3.6.2, the rank of $\overline{\mathcal{L}}(x,\vartheta)$ is also n, so $\mathrm{rank}(\overline{\mathcal{L}}(0,\lambda)) = n$ and $\overline{\mathcal{L}}(x,\vartheta)$ is simple. Note that the regular formal solution space of the original system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ is a subspace of that of $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$. Nevertheless, depending on the elementary row operations performed on $\mathcal{L}(x,\vartheta)$, it may be that the two systems share the same regular formal solution space.

The steps of the following algorithm are very close to those of Algorithm **LinSubs** developed in the previous section. The main changes are the following:

- We work with the rows of the matrix differential operator instead of working with its columns. In particular, we act on the left and compute left minimal bases;
- We consider all the rows of the left minimal bases and not only the constant ones. Note that a consequence is that the termination criterion of the algorithm slightly changes;

- As our goal is to find the regular solutions of the non-simple system, at each step of the reduction, we look at the type of the elementary row operation performed. If we apply an elementary row operation of the first and not of the second type, then we keep the index of the corresponding row in a set $\mathcal{K}$ (see Algorithm **EG_DV** below) which will be used later in Subsection 3.6.3 to reconstruct the regular solutions of the original system.

ALGORITHM **EG_DV**

INPUT: A non-simple matrix differential operator $\mathcal{L}(x,\vartheta) \in \mathbb{K}[x][\vartheta]^{n\times n}$ with invertible leading coefficient matrix.
OUTPUT: A simple matrix differential operator $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{P}(x,\vartheta)\,\mathcal{L}(x,\vartheta)$ and a set $\mathcal{K}$.

INITIALIZATION: $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)$ and $\mathcal{K} = \{\,\}$;

While $\overline{\mathcal{L}}(0,\vartheta)$ is singular **do**
1. Compute a matrix $\mathcal{B}$ whose rows form a left minimal basis of $\overline{\mathcal{L}}(0,\vartheta)$;
2. **For** each row $\mathcal{B}(k,.)$ **do**
   2.1. Let $J_k = \{j \in \{1,\ldots,n\}$ such that $\mathcal{B}(k,j) \neq 0\}$;
   2.2. Choose $i_k \in J_k$ such that $\deg_x(\overline{\mathcal{L}}(x,\vartheta)(i_k,.)) \geq \deg_x(\overline{\mathcal{L}}(x,\vartheta)(i,.))\ \forall\, i \in J_k$;
   2.3. Apply the following reduction procedure:
      2.3.(a). Let $\overline{\mathcal{L}}(x,\vartheta)(i_k,.) = \mathcal{B}(k,.)\,\overline{\mathcal{L}}(x,\vartheta)$;
      2.3.(b). Let $\beta_{i_k} = v\big(\overline{\mathcal{L}}(x,\vartheta)(i_k,.)\big)$ and $\overline{\mathcal{L}}(x,\vartheta)(i_k,.) = x^{-\beta_{i_k}}\,\overline{\mathcal{L}}(x,\vartheta)(i_k,.)$;
   2.4. **If** $\deg(\mathcal{B}(k,i_k)) \neq 0$ **then** $\mathcal{K} = \mathcal{K} \cup \{i_k\}$ **end if**;
   2.5. **If** $\deg(\mathcal{B}(k,.)) = 0$ **then** use $\mathcal{B}(k,i_k)$ as a pivot to eliminate all $\mathcal{B}(j,i_k)$ with $j \neq k$ **else** go back to Step 1 **end if**;
   end do;
end do;
Return $\overline{\mathcal{L}}(x,\vartheta)$ and $\mathcal{K}$;

Proposition 3.6.2. *Let $\mathcal{L}(x,\vartheta) = \sum_{i=0}^{\ell} A_i(x)\,\vartheta^i \in \mathbb{K}[x][\vartheta]^{n\times n}$ be a non-simple matrix differential operator with invertible leading coefficient matrix $A_\ell(x)$ and let $N = \max_{i=0,\ldots,\ell}\deg(A_i(x))$. Let ℓ_{simple} ($\ell_{\text{simple}} \leq n^{n\,N}\,\ell$) denote the order of the operator returned by Algorithm* **EG_DV**. *Then, Algorithm* **EG_DV** *stops after at most $n\,N$ calls of the reduction procedure and uses at most $O^{\sim}\big(n^{\omega+2}\,N\,\ell_{\text{simple}} + n^4\,N^2\,\ell_{\text{simple}}\big)$ arithmetic operations in $\mathbb{K}$.*

Proof. Each time we execute the reduction procedure, the degree in x of one row of the operator $\overline{\mathcal{L}}(x,\vartheta)$ decreases by at least 1 while the degrees of the other rows remain unchanged (adapt the proof of Proposition 3.5.1: the columns of the leading coefficient matrix are replaced by the rows of the matrix differential operator). Consequently, either the algorithm stops before performing $n\,N$ times the reduction procedure or after the $(n\,N)$th reduction, in which case, the output operator has constant coefficient matrices and is of rank n (by Lemma 3.6.2); therefore, it is necessarily simple. This proves the first claim. We now study the arithmetic complexity of the algorithm. We recall first that the degrees of the elements of a left minimal basis of an $n \times n$ singular matrix polynomial of rank r and degree d are bounded by $r\,d < n\,d$ (see Section 1.7 of Chapter 1). Consequently, if we suppose that, after running i times the **While** loop, the operator $\overline{\mathcal{L}}(x,\vartheta)$ is still non-simple and if $\ell_i \in \mathbb{N}^*$ ($\ell_0 = \ell$) denotes the order of $\overline{\mathcal{L}}(x,\vartheta)$ and r_i denotes the rank of $\overline{\mathcal{L}}(0,\vartheta)$, then we have $\ell_i \leq \ell_{i+1} \leq \ell_i + r_i\,\ell_i \leq n\,\ell_i$. Now in the worst case, we run $n\,N$ times the **While** loop, so if ℓ_{simple} denotes the order of the output operator then we have $\ell_{\text{simple}} \leq n^{n\,N}\,\ell$. At the ith passage in the **While** loop, Step 1 can be done using at most $O^{\sim}\big(n^{\omega+1}\,\ell_{i-1}\big)$ operations in $\mathbb{K}$ (see Lemma 1.7.2 of Chapter 1). Taking into account the

degrees of the elements of the computed left minimal basis, Step 2.3.(a) at the ith passage in the **While** loop can be performed using at most $O^{\sim}\left(n^3\,N\,\ell_{i-1}\right)$ operations in $\mathbb{K}$[3] and Step 2.5 in $O\left(n^3\,\ell_{i-1}\right)$ operations in $\mathbb{K}$. Hence Algorithm **EG_DV** returns a simple operator after at most $O^{\sim}\left(n^{\omega+2}\,N\,\ell_{\text{simple}}+n^4\,N^2\,\ell_{\text{simple}}\right)$ operations in $\mathbb{K}$. □

We will make a few comments on Algorithm **EG_DV**:

1. We use $\mathcal{B}(k,i_k)$ as a pivot to eliminate all the elements $\mathcal{B}(j,i_k)$ with $j\neq k$ only if $\mathcal{B}(k,.)$ is a constant row, otherwise we may increase the degrees of the elements of $\mathcal{B}$ and consequently, we cannot ensure in the proof of Proposition 3.6.2 that $\ell_{i+1}\leq\ell_i+r_i\,\ell_i$;

2. Algorithm **EG_DV** can be applied more generally to any non-simple matrix differential operator $\mathcal{L}(x,\vartheta)=\sum_{i=0}^{\ell}A_i(x)\,\vartheta^i\in\mathbb{K}[x][\vartheta]^{n\times n}$ of rank n and not necessarily with invertible leading coefficient $A_\ell(x)$;

3. The complexity result that we give is a worst case estimate. In practice, the potentially exponential growth of the order of the operator does not seem to be a serious limitation.

Example 3.6.1. *Consider the matrix differential operator defined by*

$$\mathcal{L}(x,\vartheta)=\begin{pmatrix}x&0&0\\0&x&x\\0&0&1\end{pmatrix}\vartheta^3+\begin{pmatrix}0&x^3&1\\0&x^2&1\\x&1&0\end{pmatrix}\vartheta^2+\begin{pmatrix}0&1&x^2\\0&1&x\\1&0&2\,x\end{pmatrix}\vartheta+\begin{pmatrix}1&0&0\\1&0&0\\0&x^2&4\,x^3\end{pmatrix}.$$

Its associated matrix polynomial $\mathcal{L}(0,\vartheta)$ given by

$$\mathcal{L}(0,\vartheta)=\begin{pmatrix}1&\vartheta&\vartheta^2\\1&\vartheta&\vartheta^2\\\vartheta&\vartheta^2&\vartheta^3\end{pmatrix}$$

is singular; thus the operator $\mathcal{L}(x,\vartheta)$ is non-simple. A right minimal basis of $\mathcal{L}(0,\vartheta)$ is given by the columns of the matrix

$$\begin{pmatrix}-\vartheta&0\\1&-\vartheta\\0&1\end{pmatrix};$$

hence, according to Theorem 3.5.1, there exists no linear substitution yielding a simple operator. Consequently, we will apply Algorithm **EG_DV** *above. A left minimal basis of $\mathcal{L}(0,\vartheta)$ is given by the rows of the matrix*

$$\mathcal{B}=\begin{pmatrix}-1&1&0\\\vartheta&0&-1\end{pmatrix}.$$

Let us start by considering the first row $\mathcal{B}(1,.)=\begin{pmatrix}-1&1&0\end{pmatrix}$ whose first two components are nonzero. Since $\deg_x\left(\mathcal{L}(x,\vartheta)(1,.)\right)=3>\deg_x\left(\mathcal{L}(x,\vartheta)(2,.)\right)=2$, we will reduce the first row of $\mathcal{L}(x,\vartheta)$, i.e., *$i_1=1$. Replacing $\mathcal{L}(x,\vartheta)(1,.)$ by $\mathcal{B}(1,.)\mathcal{L}(x,\vartheta)=-\mathcal{L}(x,\vartheta)(1,.)+\mathcal{L}(x,\vartheta)(2,.)$, we get the matrix differential operator*

$$\mathcal{L}_1(x,\vartheta)=\begin{pmatrix}-x&x&x\\0&x&x\\0&0&1\end{pmatrix}\vartheta^3+\begin{pmatrix}0&x^2-x^3&0\\0&x^2&1\\x&1&0\end{pmatrix}\vartheta^2+\begin{pmatrix}0&0&x-x^2\\0&1&x\\1&0&2\,x\end{pmatrix}\vartheta+\begin{pmatrix}0&0&0\\1&0&0\\0&x^2&4\,x^3\end{pmatrix}.$$

[3] We suppose that the Fast Fourier Transform can be used so that two univariate polynomials with coefficients in $\mathbb{K}$ and degree bounded by d can be multiplied in $O^{\sim}(d)$ arithmetic operations in $\mathbb{K}$, see [31].

Now $\beta_1 = v(\mathcal{L}_1(x,\vartheta)(1,.)) = 1$ *so we multiply* $\mathcal{L}_1(x,\vartheta)(1,.)$ *on the left by* x^{-1}*. We obtain the operator*

$$\mathcal{L}_2(x,\vartheta) = \begin{pmatrix} -1 & 1 & 1 \\ 0 & x & x \\ 0 & 0 & 1 \end{pmatrix} \vartheta^3 + \begin{pmatrix} 0 & x-x^2 & 0 \\ 0 & x^2 & 1 \\ x & 1 & 0 \end{pmatrix} \vartheta^2 + \begin{pmatrix} 0 & 0 & 1-x \\ 0 & 1 & x \\ 1 & 0 & 2\,x \end{pmatrix} \vartheta + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & x^2 & 4\,x^3 \end{pmatrix}$$

which is left-equivalent to $\mathcal{L}(x,\vartheta)$ *since* $\deg(\mathcal{B}(1,1)) = 0$*. We now use* $\mathcal{B}(1,1) = -1$ *as a pivot to eliminate* $\mathcal{B}(2,1)$*. Consequently,* $\mathcal{B}$ *becomes*

$$\mathcal{B}_1 = \begin{pmatrix} -1 & 1 & 0 \\ 0 & \vartheta & -1 \end{pmatrix}.$$

Let us now consider the second row $\mathcal{B}_1(2,.)$ *which is in the left minimal basis of* $\mathcal{L}_2(0,\vartheta)$*. Since* $\deg_x(\mathcal{L}_2(x,\vartheta)(3,.)) = 3 > \deg_x(\mathcal{L}_2(x,\vartheta)(2,.)) = 2$*, we have* $i_2 = 3$*; we replace* $\mathcal{L}_2(x,\vartheta)(3,.)$ *by* $\mathcal{B}_1(2,.)\mathcal{L}_2(x,\vartheta) = -\mathcal{L}_2(x,\vartheta)(3,.) + \vartheta\,\mathcal{L}_2(x,\vartheta)(2,.)$ *and obtain the operator of order* 4

$$\mathcal{L}_3(x,\vartheta) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & x & x \end{pmatrix} \vartheta^4 + \begin{pmatrix} -1 & 1 & 1 \\ 0 & x & x \\ 0 & x+x^2 & x \end{pmatrix} \vartheta^3 + \begin{pmatrix} 0 & x-x^2 & 0 \\ 0 & x^2 & 1 \\ -x & 2\,x^2 & x \end{pmatrix} \vartheta^2 + \begin{pmatrix} 0 & 0 & 1-x \\ 0 & 1 & x \\ 0 & 0 & -x \end{pmatrix} \vartheta + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -x^2 & -4\,x^3 \end{pmatrix}.$$

Since $\deg(\mathcal{B}_1(2,3)) = 0$*,* $\mathcal{L}_3(x,\vartheta)$ *is also left-equivalent to* $\mathcal{L}(x,\vartheta)$*. Now* $\beta_3 = v(\mathcal{L}_3(x,\vartheta)(3,.)) = 1$*, so we multiply the third row of* $\mathcal{L}_3(x,\vartheta)$ *on the left by* x^{-1} *and get*

$$\mathcal{L}_4(x,\vartheta) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \vartheta^4 + \begin{pmatrix} -1 & 1 & 1 \\ 0 & x & x \\ 0 & 1+x & 1 \end{pmatrix} \vartheta^3 + \begin{pmatrix} 0 & x-x^2 & 0 \\ 0 & x^2 & 1 \\ -1 & 2\,x & 1 \end{pmatrix} \vartheta^2 + \begin{pmatrix} 0 & 0 & 1-x \\ 0 & 1 & x \\ 0 & 0 & -1 \end{pmatrix} \vartheta + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -x & -4\,x^2 \end{pmatrix}.$$

One can easily check that the matrix polynomial associated with $\mathcal{L}_4(x,\vartheta)$ *given by*

$$\mathcal{L}_4(0,\lambda) = \begin{pmatrix} -\lambda^3 & \lambda^3 & \lambda^3+\lambda \\ 1 & \lambda & \lambda^2 \\ -\lambda^2 & \lambda^4+\lambda^3 & \lambda^4+\lambda^3+\lambda^2-\lambda \end{pmatrix}$$

is regular and hence the operator $\mathcal{L}_4(x,\vartheta)$ *is simple. Furthermore, it is left-equivalent to* $\mathcal{L}(x,\vartheta)$*. Consequently, the regular formal solution spaces of* $\mathcal{L}(x,\vartheta)(y(x)) = 0$ *and* $\mathcal{L}_4(x,\vartheta)(y(x)) = 0$ *are exactly the same, so applying Algorithm* **BCE_V2** *to* $\mathcal{L}_4(x,\vartheta)$ *yields the general regular formal solution of* $\mathcal{L}(x,\vartheta)$*.* ■

Example 3.6.2. *Consider the matrix differential operator* $\mathcal{L}(x,\vartheta)$ *given by*

$$\mathcal{L}(x,\vartheta) = \begin{pmatrix} 1 & 0 \\ 0 & x \end{pmatrix} \vartheta^2 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \vartheta + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.17}$$

Its associated matrix polynomial given by

$$\mathcal{L}(0,\vartheta) = \begin{pmatrix} \vartheta^2 & \vartheta \\ \vartheta & 1 \end{pmatrix}$$

is singular and hence the operator $\mathcal{L}(x,\vartheta)$ is non-simple. According to Proposition 3.4.1, a simple operator cannot be obtained from $\mathcal{L}(x,\vartheta)$ by means of a linear substitution. Consequently, we will apply Algorithm **EG_DV** *to $\mathcal{L}(x,\vartheta)$. A left minimal basis of $\mathcal{L}(0,\vartheta)$ is composed of one vector $v = \begin{pmatrix}-1 & \vartheta\end{pmatrix}$. Since $\deg_x(\mathcal{L}(x,\vartheta)(2,.)) > \deg_x(\mathcal{L}(x,\vartheta)(1,.))$, we replace $\mathcal{L}(x,\vartheta)(2,.)$ by $v\,\mathcal{L}(x,\vartheta) = \vartheta\,\mathcal{L}(x,\vartheta)(2,.) - \mathcal{L}(x,\vartheta)(1,.)$; this is a combination of two elementary row operations of the first type. It yields the matrix differential operator*

$$\mathcal{L}_1(x,\vartheta) = \begin{pmatrix}0 & 0\\ 0 & x\end{pmatrix}\vartheta^3 + \begin{pmatrix}1 & 0\\ 0 & x\end{pmatrix}\vartheta^2 + \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}\vartheta$$

which is not left-equivalent to $\mathcal{L}(x,\vartheta)$. So we set $\mathcal{K} = \{2\}$. Finally, we multiply the second row of $\mathcal{L}_1(x,\vartheta)$ on the left by x^{-1} and we get

$$\mathcal{L}_2(x,\vartheta) = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}\vartheta^3 + \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\vartheta^2 + \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}\vartheta. \tag{3.18}$$

The latter system is simple but not left-equivalent to $\mathcal{L}(x,\vartheta)$ given by (3.17). Hence the regular formal solution space of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ is a subspace of that of $\mathcal{L}_2(x,\vartheta)(y(x)) = 0$. ■

3.6.3 Reconstruction of the regular solutions

Now, we will explain how to reconstruct the general regular solution up to order $\nu \in \mathbb{N}$ of a non-simple system $\mathcal{L}(x,\vartheta)(y(x)) = 0$ from that of the output of Algorithm **EG_DV**.

Two cases have to be considered. If the output operator $\overline{\mathcal{L}}(x,\vartheta)$ of Algorithm **EG_DV** is left-equivalent to the input one $\mathcal{L}(x,\vartheta)$ (this corresponds to $\mathcal{K} = \{\,\}$), then the general regular solution of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ is exactly that of $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$. Consequently, to get the general regular solution of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ up to order ν, it suffices to compute that of $\overline{\mathcal{L}}(x,\vartheta)(y(x)) = 0$ up to order ν by applying Algorithm **BCE_V2**.

Otherwise, *i.e.*, if $\mathcal{K}$ is a nonempty set, $\overline{\mathcal{L}}(x,\vartheta)$ is not left-equivalent to $\mathcal{L}(x,\vartheta)$ and we can proceed as follows. First, we compute the general regular solution $z(x)$ of system $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ up to order ν by applying Algorithm **BCE_V2**. Write $z(x) = \sum_{i=1}^{r} x^{\lambda_i}\, z_i(x)$, where $x^{\lambda_i}\, z_i(x)$ is the general regular solution associated with the set σ_i, computed up to order ν (see Subsection 2.6.3 of Chapter 2). Then, we consider the subsystem of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ formed by the equations given by the rows of $\mathcal{L}(x,\vartheta)$ of indices $j \in \mathcal{K}$ and we plug $z(x)$ into it. Hence, $z(x)$ is a general regular solution up to order ν of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ if and only if the coefficients of x^{λ_i+k} in $\mathcal{L}(x,\vartheta)(j,.)(z(x))$ for $j \in \mathcal{K}$, $i = 1,\ldots,r$ and $k = 0,\ldots,\nu$ are all equal to zero (see Remark 2.6.3 of Chapter 2). This yields a system of linear equations in the parameters appearing in $z(x)$. Finally, solving this system and substituting its solution into $z(x)$, we get the general regular solution of $\mathcal{L}(x,\vartheta)(y(x)) = 0$ up to order ν.

Example 3.6.3. *We are interested in computing the general regular solution of the linear differential system $\mathcal{L}(x,\vartheta)((y(x)) = 0$, where $\mathcal{L}(x,\vartheta)$ is given by (3.17). As we have already seen in Example 3.6.2, Algorithm* **EG_DV** *applied to $\mathcal{L}(x,\vartheta)$ returns a non left-equivalent operator $\mathcal{L}_2(x,\vartheta)$ given by (3.18) and the set $\mathcal{K} = \{2\}$. Since the system $\mathcal{L}_2(x,\vartheta)((y(x)) = 0$ has constant coefficient matrices, its regular solutions are then of the form given by Lemma 2.4.1 of Chapter 2. Since $\sigma(\mathcal{L}_2(0,\lambda)) = \{-1,0\}$, the general regular solution of $\mathcal{L}_2(x,\vartheta)(y(x)) = 0$ can be written as $x^{-1}\,(U_0 + U_1\,x)$, where U_0 and U_1 are two vector polynomials in $\log(x)$ whose coefficients depend on arbitrary constants. Consequently, Algorithm* **BCE_V2** *applied to $\mathcal{L}_2(x,\vartheta)$ with $\nu = 1$ returns the general regular solution of $\mathcal{L}_2(x,\vartheta)(y(x)) = 0$ given by*

$$z(x) = \begin{pmatrix} C_1\,x^{-1} + C_2 + C_3\,\log(x) - \frac{C_4}{2}\,\log^2(x)\\ C_1\,x^{-1} + C_5 + C_4\,\log(x)\end{pmatrix},$$

where the C_i's are arbitrary constants in $\mathbb{Q}$. Since $\mathcal{K} = \{2\}$, we plug $z(x)$ into the second equation of the original system and we find

$$\mathcal{L}(x, \vartheta)(2, .)(z(x)) = C_1 + C_3 + C_5.$$

Therefore, the vector z is the general regular solution of $\mathcal{L}(x, \vartheta)(y(x)) = 0$ if and only if the constants C_1, C_3 and C_5 satisfies $C_1 = -C_3 - C_5$. Hence, the general regular solution of system $\mathcal{L}(x, \vartheta)(y(x)) = 0$ is given by

$$\begin{pmatrix} C_2 + C_3\,(\log(x) - x^{-1}) - \frac{C_4}{2}\,\log^2(x) - C_5\,x^{-1} \\ -C_3\,x^{-1} + C_5\,(1 - x^{-1}) + C_4\,\log(x) \end{pmatrix}.$$

■

Chapter 4

On k-Simple Forms of First-Order Linear Differential Systems and their Computation

Contents

4.1 Introduction and motivation

Consider a system of first-order linear differential equations of the form

$$\vartheta(y(x)) = A(x)\,y(x), \quad \text{with } A(x) = \frac{1}{x^p}\sum_{i=0}^{\infty} A_i\,x^i, \tag{4.1}$$

where $A_i \in \mathbb{K}^{n\times n}$ such that $A_0 \neq 0$ and $p \in \mathbb{N}$ is the Poincaré-rank of the system. It is well-known (see [9, 95, 101]) that System (4.1) has n linearly independent formal solutions of the form

$$y_i(x) = \exp\Big(q_i\left(x^{-1/r_i}\right)\Big)\,x^{\lambda_i}\,z_i\left(x^{1/r_i}\right), \quad \text{for } i = 1, \ldots, n, \tag{4.2}$$

where $r_i \in \mathbb{N}^*$, $q_i(t) \in t\,\overline{\mathbb{K}}[t]$, $\lambda_i \in \overline{\mathbb{K}}$ and $z_i\big(x^{1/r_i}\big) \in \overline{\mathbb{K}}[[x^{1/r_i}]][\log(x)]^n$. For some integer $i \in \{1, \ldots, n\}$, if one has $q_i = 0$, then r_i can be chosen equal to 1 and $y_i(x)$ is a regular solution of System (4.1). Otherwise, *i.e.*, if one has $q_i \neq 0$, $y_i(x)$ is called an *irregular solution*. When System (4.1) admits at least one irregular solution, the point $x = 0$ is said to be an *irregular singularity* for the system. Remark that each solution y_i can be written as

$$y_i = \exp\left(\int w_i\right) z_i \quad \text{with} \quad w_i = \frac{dq_i}{dx} + \frac{\lambda_i}{x} \in x^{-1}\overline{\mathbb{K}}[x^{-1/r_i}]$$

and the polynomials q_i are invariant with respect to any gauge transformation $y(x) = T(x)\,z(x)$ (see Subsection 2.2.1 of Chapter 2) with $T(x) \in \mathrm{GL}_n\Big(\overline{\mathbb{K}((x))}\Big)$.

A classical approach for computing the formal solutions (4.2) of System (4.1) consists in reducing the system, by means of cyclic vectors [11, 35] for example, to an nth order scalar linear differential equation. The classification of the singularity can then be read off directly from the *Newton polygon* [13, 94] which is constructed from the valuations of the coefficients of the scalar equation at $x = 0$; the singularity is regular if and only if the Newton polygon has only one slope equal to 0 (see [94, Chap. 2] and references therein). When the singularity is irregular, the positive slopes of the Newton polygon give the "degrees" in $1/x$ of the polynomials q_i occurring in the exponential parts of the solutions (4.2) and the roots of the *Newton polynomials*[1] give the leading coefficients of these q_i's (see [13, 26, 94]). The *Newton algorithm* [10, 26, 94] can then be used to compute a fundamental system of formal solutions of the scalar equation from which the solutions of the given system can be deduced. However, one would like to avoid the computation of an equivalent scalar linear differential equation as it can be very costly in general (see [36, Chap. 1]).

The Newton polygon of a system of the form (4.1) is defined as the Newton polygon of any nth order linear differential equation equivalent to (4.1). Unfortunately, the Newton polygon of (4.1) cannot immediately be read off from the system matrix $A(x)$. Nevertheless, there exist direct methods (see [13, 33, 85, 101] and references therein) for constructing the solutions (4.2) of (4.1) without converting into an nth order linear differential equation. These methods determine first the polynomials q_i. After this, the problem can then be reduced to solving regular formal solutions of systems of the form (4.1). These algorithms are essentially based on the *formal reduction* procedure: reducing the problem of size n into several independent subproblems of smaller size (*Splitting Lemma* [13, Th. 2] and *Generalized Splitting Lemma* [85, Prop. 3.2]). Here, we will be interested in particular in the algorithm developed in [85] which is able to detect and separate the polynomials q_i having integer degrees from those having non integer degrees for which the task seems to be a bit more complicated. Let k be a nonnegative integer smaller than or equal to the Poincaré-rank p of System (4.1). Write System (4.1) in the form

$$\mathcal{D}_k(y(x)) = D(x)\,\vartheta_k(y(x)) + N(x)\,y(x) = 0, \tag{4.3}$$

where $\vartheta_k = x^k\,\vartheta$, $D(x)$ and $N(x)$ belong to $\mathbb{K}[[x]]^{n\times n}$ such that $D(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ and for $i = 1, \ldots, n$, $\min\{v(D(i,.)), v(N(i,.)\} = 0$ (see [84, 85] for more details). The action of $\mathcal{D}_k$ on a solution of the form $y = \exp\big(\int w\big)\,z$ with $w \in \overline{\mathbb{K}((x))}$ of the form

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots,$$

and $z \in \overline{\mathbb{K}}[[x^{\frac{1}{r}}]]^n$ $(r \in \mathbb{N}^*)$ such that $z(0) \neq 0$ gives

$$\exp\left(-\int w\right)\mathcal{D}_k(y) = (D(0)\,\lambda_0 + N(0))\,z(0) + \cdots,$$

where the dots stand for terms of higher valuation. Therefore, $y = \exp\big(\int w\big)\,z$ is a solution of $\mathcal{D}_k(y) = 0$ implies that $(D(0)\,\lambda_0 + N(0))\,z(0) = 0$. Consequently, if the matrix pencil defined by

$$L_k(\lambda) = D(0)\,\lambda + N(0)$$

is singular, then no useful information is provided. Otherwise, λ_0 has to be chosen as an eigenvalue of $L_k(\lambda)$ and $z(0)$ as an eigenvector of $L_k(\lambda)$ associated with λ_0 (see Section 1.1 of Chapter 1). Systems of the form (4.3) with a regular matrix pencil $L_k(\lambda)$ are called *k-simple systems* (see Definition 4.2.1 below). Thus, the notion of 0-simple systems coincides with that

[1] Newton's polynomials are also known as *characteristic polynomials*. They are polynomials associated with the slopes of the Newton polygon.

of simple systems viewed in Definition 3.2.1 of the previous chapter. It has been shown in [85, Th. 3.3] that if System (4.1) can be written as a k-simple system (4.3) and if λ_0 is an eigenvalue of the matrix pencil $L_k(\lambda)$ of algebraic multiplicity m, then there exist m linearly independent formal solutions of (4.1) of the form $y(x) = \exp\big(\int w\big)\, z(x)$, where $w \in x^{-1}\overline{\mathbb{K}}[x^{-1/r}]$ $(r \in \mathbb{N}^*)$ can be written as

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots,$$

here the dots stand for terms of higher valuation, and $z \in \overline{\mathbb{K}}[[x^{1/r}]][\log(x)]^n$. We see then that the determinant of $L_k(\lambda)$ plays the same role as a Newton polynomial. Thus, assuming that for $k = 0, 1, \ldots, p$, the matrix pencil $L_k(\lambda)$ is regular, the Newton polygon of System (4.1) has

- a zero slope if and only if $L_0(\lambda)$ has at least one eigenvalue, *i.e.*, $\det(L_0(\lambda)) \notin \mathbb{K}$, and
- an integer slope k between 1 and p if and only if $L_k(\lambda)$ has a nonzero eigenvalue.

Thus, to determine whether an integer $k \in \{0, \ldots, p\}$ is a slope of the Newton polygon of (4.1), the latter system needs first to be written as k-simple system of the form (4.3). But this is not always trivial since the condition on the regularity of the matrix pencil $L_k(\lambda)$ is sometimes hard to satisfy. However, it has been shown in [85, Th. 4.1] that if System (4.1) is in the so-called *super-reduced* form [58, 24, 25], then it can be written as k-simple system for all the integers k between 0 and p.

In this chapter, we consider the class of systems of the form (4.3) with $k \in \mathbb{N}$. Motivated by the fact that for some integer k, a k-simple system (4.3) written as

$$\vartheta(y(x)) = -x^{-k}\, D^{-1}(x)\, N(x)\, y(x)$$

is not necessarily super-reduced, the purpose of this chapter is the development of a direct method transforming a given system of the form (4.3) into a k-simple equivalent one (see Definition 4.2.2 below). Thus, we will be able in particular to answer the question whether a fixed integer k is a slope of the Newton polygon of System (4.1) without reducing the latter system into a super-reduced form. Our approach is also useful when one is looking for regular solutions of System (4.1), or more generally, of systems of the form

$$D(x)\, \vartheta(y(x)) + N(x)\, y(x) = 0 \tag{4.4}$$

with $D(x)$ and $N(x)$ in $\mathbb{K}[[x]]^{n\times n}$ and $D(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$: indeed, the algorithms that we have developed in Chapter 2 and the algorithm of [21] for computing regular solutions are both built on the prerequisite condition that System (4.4) is simple.

The chapter is organized as follows. First, in Section 4.2, we review the notion of k-simple systems which has been introduced in [84, 85] as a generalization of that of simple systems, see Chapter 3. Then, we recall the notion of super-reduction which consists a standard approach for computing k-simple forms. In Section 4.3, we develop our direct method for computing a k-simple system equivalent to (4.3). We illustrate our approach with examples and we study its arithmetic complexity in Section 4.4. In Section 4.5, we highlight a characteristic of our algorithm. Finally, we provide in Section 4.6 an example recapitulating all the notions viewed in this chapter.

Notation. Throughout this chapter, for a matrix M with entries in $\mathbb{K}[[x]]$, the notation M_0 denotes matrix M evaluated at $x = 0$ and $M_{|\nu}$ denotes matrix M truncated at order $\nu \in \mathbb{N}$. For $f \in \overline{\mathbb{K}((x))} \setminus \{0\}$, $v(f)$ stands for the x-adic valuation of f and $\ell c(f)$ for the leading coefficient of f, *i.e.*, the coefficient of $x^{v(f)}$ in f. By convention, we take $v(0) = +\infty$. For an n-dimensional

multi-index $\alpha = (\alpha_1, \ldots, \alpha_n)$, $|\alpha|$ denotes the sum of its components, that is, $|\alpha| = \sum_{i=1}^{n} \alpha_i$. We recall that the Kronecker delta $\delta_{i,j}$ is defined by

$$\delta_{i,j} = \begin{cases} 1 & \text{if} \quad i = j, \\ 0 & \text{if} \quad i \neq j. \end{cases}$$

Finally, for a nonnegative integer k, ϑ_k stands for the derivation $x^k\vartheta$ and the symbol $\int$ denotes the integration with respect to dx, *i.e.*,

$$\text{for } w \in \overline{\mathbb{K}((x))} = \bigcup_{r\geq 1} \overline{\mathbb{K}}((x^{\frac{1}{r}})), \quad \frac{d}{dx}\left(\int w\right) = w.$$

4.2 On k-simple linear differential systems and the super-reduction

4.2.1 Definitions

Definition 4.2.1 ([85], Def. 3.3)**.** *Let $k \in \mathbb{N}$. A linear differential system of the form*

$$\mathcal{D}_k(y(x)) = D(x)\,\vartheta_k(y(x)) + N(x)\,y(x) = 0,$$

or the matrix differential operator $\mathcal{D}_k = D(x)\,\vartheta_k + N(x)$, with $D(x)$, $N(x) \in \mathbb{K}[[x]]^{n\times n}$ such that $D(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ and for $i = 1, \ldots, n$, $\min\{v(D(i,.)), v(N(i,.)\} = 0$ is said to be k-simple*, or* simple with respect to ϑ_k*, or just* simple *when no confusion arises, if the matrix pencil $L_k(\lambda)$ defined by*

$$L_k(\lambda) = D_0\,\lambda + N_0$$

is regular. In the sequel, we will refer to $L_k(\lambda)$ as the matrix pencil associated with $\mathcal{D}_k$*.*

Definition 4.2.2. *Two matrix differential operators $\mathcal{D}_k = D(x)\,\vartheta_k + N(x)$ and $\widetilde{\mathcal{D}}_k = \widetilde{D}(x)\,\vartheta_k + \widetilde{N}(x)$, with matrices $D(x)$, $N(x)$, $\widetilde{D}(x)$ and $\widetilde{N}(x)$ belonging to $\mathbb{K}((x))^{n\times n}$, are said to be* equivalent *if there exist two matrices $S(x)$ and $T(x)$ in $\mathrm{GL}_n(\mathbb{K}((x)))$ such that $\widetilde{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,T(x)$, that is, $\widetilde{D}(x) = S(x)\,D(x)\,T(x)$ and $\widetilde{N}(x) = S(x)\,N(x)\,T(x) + S(x)\,D(x)\,\vartheta_k(T(x))$.*

It is important to point out that the multiplication of an operator $\mathcal{D}_k \in \mathbb{K}[[x]][\vartheta_k]^{n\times n}$ on the left and on the right by unimodular matrices of $\mathbb{K}[[x]]^{n\times n}$, *i.e.*, by invertible matrices over $\mathbb{K}[[x]]$, preserves the k-simplicity of the operator. In other terms, if $S(x)$ and $T(x)$ are two unimodular matrices of $\mathbb{K}[[x]]^{n\times n}$, then two equivalent operators $\mathcal{D}_k$ and $\widetilde{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,T(x) = \widetilde{D}(x)\,\vartheta + \widetilde{N}(x)$ are simultaneously either simple or non-simple with respect to ϑ_k. Indeed, let $\widetilde{L}_k(\lambda)$ denote the matrix pencil associated with $\widetilde{\mathcal{D}}_k$. By definition, $\widetilde{L}_k(\lambda)$ is equal to

$$\begin{aligned}
\widetilde{L}_k(\lambda) &= \widetilde{D}(x)\,\lambda + \widetilde{N}(x)\,\Big|_{x=0} \\
&= S(x)\left(D(x)\,\lambda + N(x) + D(x)\,\vartheta_k(T(x))T^{-1}(x)\right)T(x)\,\Big|_{x=0} \\
&= S_0\,(D_0\,\lambda + N_0)\,T_0 \qquad \left(\text{since } v(\vartheta_k(T(x))) \geq k+1 \text{ and } v\big(T^{-1}(x)\big) \geq 0\right) \\
&= S_0\,L_k(\lambda)\,T_0
\end{aligned}$$

with S_0 and T_0 both invertible. Thus, $\widetilde{L}_k(\lambda)$ is regular if and only if $L_k(\lambda)$ is so.

4.2.2 The notion of super-reduction

In this section, we review the notion of *super-reduction* and the connexion between this notion and that of k-simplicity. We start by the reduction in the sense of Moser since it is the base of the super-reduction.

Let $m(A)$ and $\mu(A)$ be two rational numbers associated with System (4.1) and defined respectively by

$$m(A) = \max\Big\{1, p + \tfrac{\operatorname{rank}(A_0)}{n}\Big\}$$
$$\text{and} \quad \mu(A) = \min\big\{m(T^{-1}(A\,T - \vartheta(T)));\, T \in \mathrm{GL}_n(\mathbb{K}((x)))\big\}\,.$$

If System (4.1) satisfies $m(A) = \mu(A)$, then it is said to be *Moser-reduced*. Otherwise, it is said to be *Moser-reducible*. In the latter case, Moser proves in [77] that there exists an invertible matrix $T(x) \in \mathbb{K}[x]^{n\times n}$ such that the system $\vartheta(z(x)) = B(x)\,z(x)$, where $B(x) = T^{-1}(x)\,A(x)\,T(x) - T^{-1}(x)\,\vartheta(T(x))$, satisfies $m(B) = \mu(A)$. To construct such a transformation $T(x)$, Moser proposes an algorithm which has been later improved in [56, 12].

Apart from the classification of the singularity of (4.1) (see Subsection 2.2.1 of Chapter 2), Moser's algorithm answers the question whether the Katz invariant κ of System (4.1), which is the biggest slope of the Newton Polygon, is an integer or not, see [13]. Indeed, suppose that System (4.1) is Moser-reduced and that its Poincaré-rank p is greater than or equal to 1. The Katz invariant of (4.1) is an integer if and only if the matrix A_0 is non nilpotent. In this case, we have $\kappa = p$. When $\kappa \in \mathbb{Q} \setminus \mathbb{N}$, the matrix A_0 is necessarily nilpotent and p is the smallest integer greater than κ, *i.e.*, $p - 1 < \kappa < p$.

Starting from the notion of Moser-reduction, Hilali & Wazner introduced in [58] the notion of super-reduction. As we will see in Lemma 4.2.1 below, the super-reduction provides information on the integer slopes of the Newton polygon of System (4.1). Furthermore, it has been shown that the invariants of Malgrange and of Gerard-Levelt [55] can be directly read off from a super-reduced form. Following the presentation in [14], we will define the super-reduction by working with the rows of the system matrix $A(x)$. For $0 \le i \le p-1$, let n_i denote the number of rows of $A(x)$ with valuation $i - p$. For $1 \le k \le p$, define the rational numbers $m_k(A)$ by

$$\begin{cases} m_k(A) = 1 & \text{if } p = 0, \\ m_k(A) = p + \dfrac{n_0}{n} + \dfrac{n_1}{n^2} + \cdots + \dfrac{n_{k-1}}{n^k} & \text{if } p \ge 1 \end{cases}$$

and $\mu_k(A)$ by

$$\mu_k(A) = \min\big\{m_k\left(T^{-1}(A\,T - \vartheta(T))\right);\, T \in \mathrm{GL}_n(\mathbb{K}((x)))\big\}\,.$$

System (4.1) is said to be *k-reduced* if $m_k(A) = \mu_k(A)$ and *super-reduced* if it is p-reduced. The notion of 1-reduced systems coincides then with that of Moser-reduced systems.

In [58], Hilali and Wazner provide a criterion to decide whether a system of the form (4.1) is k-reduced. For $1 \le j \le p$, define

$$r_j = j\,n_0 + (j-1)\,n_1 + \cdots + n_{j-1} \quad \text{and} \quad \theta_j(A, x, \lambda) = x^{r_j} \det\big(I_n\,\lambda - x^{p-j}A(x)\big)\,. \tag{4.5}$$

According to [58, Lemma 2.3.1], the $\theta_j(A, x, \lambda)$ are polynomials in λ with coefficients in $\mathbb{K}[[x]]$.

Proposition 4.2.1 ([58], Th. 1)**.** *Let $k \in \{1, \ldots, p\}$. The system (4.1) is k-reduced if and only if for $i = 1, \ldots, k$, $\theta_i(A, 0, \lambda)$ does not vanish identically.*

It follows that System (4.1) is k-reduced if and only if it is j-reduced for $1 \leq j \leq k$. In [58], the authors give an algorithm for computing a super-reduced system gauge-equivalent to (4.1); starting by applying Moser's algorithm to System (4.1), they explain how to lift a $(k-1)$-reduced system to a k-reduced one. Recently, another algorithm for computing a super-reduced form has been proposed by Barkatou & Pflügel in [25]. Their method reduces the computation of a super-reduced system to that of several Moser-reduced systems of smaller size using a block-reduction algorithm transforming the system into a lower block-triangular one. This algorithm computes a super-reduced system gauge-equivalent to (4.1) using $O\big(\nu\, p\, n^4 \min\{n-1, p\}\big)$ operations in $\mathbb{K}$, where ν is the number of coefficients in the expansion of the system matrix $A(x)$ of (4.1) taking into account (see [25, Prop. 4.3]).

The following lemma shows the link between super-reduced forms and k-simple systems.

Lemma 4.2.1 ([85], Th. 4.1)**.** *Any super-reduced system of the form (4.1) can be written as k-simple system for any integer $0 \leq k \leq p-1$.*

For the sake of completeness, we repeat the proof here.

Proof. System (4.1) is super-reduced implies that it is $(p-k)$-reduced for $0 \leq k \leq p-1$. Thus $\theta_{p-k}(A, 0, \lambda) \neq 0$ for $0 \leq k \leq p-1$. Define the matrix $\alpha = \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$ with $\alpha_i = \max\{0, -k - v(A(x)(i,.))\}$. Put $D(x) = x^{\alpha} \in \mathbb{K}[x]^{n\times n}$. It is easy to check that $\det(D(x)) = x^{r_{p-k}}$ with r_{p-k} given by (4.5). Therefore, we have $\theta_{p-k}(A, x, \lambda) = \det(D(x)\,\lambda - x^k D(x)\,A(x))$. From the construction of matrix $D(x)$, we have $v(D(x)A(x)) \geq -k$. Hence, the matrix $N(x) = -x^k D(x)A(x)$ belongs to $\mathbb{K}[[x]]^{n\times n}$ and $\theta_{p-k}(A, 0, \lambda) = \det(D(0)\,\lambda + N(0))$. On the other hand, multiplying System (4.1) on the left by $x^k D(x)$, we find

$$D(x)\,\vartheta_k(Y(x)) + N(x)Y(x) = 0,$$

with $\det(D(0)\,\lambda + N(0)) = \theta_{p-k}(A, 0, \lambda) \neq 0$. Hence, the latter system is k-simple. □

It is important to note that a k-simple system of the form (4.3) written as $\vartheta(y(x)) = -x^{-k} D^{-1}(x)\, N(x)\, y(x)$ is not necessarily super-reduced not even j-reduced for some positive integer j (see for example [21, Example 5]). Thus, it is worth looking for an alternative way to compute k-simple systems without using the super-reduction algorithm. This is the main purpose of this chapter and our contribution will be explained in the following section.

4.3 Direct approach for computing k-simple forms

Let k be a nonnegative integer. We consider a first-order matrix differential operator written with ϑ_k of the form

$$\mathcal{D}_k = D(x)\,\vartheta_k + N(x), \tag{4.6}$$

where $D(x)$ and $N(x)$ are two matrices of $\mathbb{K}[[x]]^{n\times n}$ such that $D(x) \in \mathrm{GL}_n(\mathbb{K}((x)))$ and for $i = 1, \ldots, n$, $\min\{v(D(i,.)), v(N(i,.))\} = 0$. We assume that $\mathcal{D}_k$ is not k-simple. We will develop a direct approach for computing a k-simple first-order matrix differential operator equivalent to $\mathcal{D}_k$ in the sense of Definition 4.2.2. This approach is based on the algebraic treatment of the matrix pencil associated with $\mathcal{D}_k$.

We introduce the set $\mathcal{A}_n$ of invertible matrices M of $\mathbb{K}[[x]]^{n\times n}$ satisfying

1. $\forall\, i = 1, \ldots, n$, $\ell c(M(i,i)) = 1$,
2. $\forall\, i = 1, \ldots, n-1$, $v(M(i,i)) \leq v(M(i+1, i+1))$,
3. $\forall\, i = 1, \ldots, n$ and $j = 1, \ldots, n$ such that $i \neq j$, $v(M(i,j)) > v(M(i,i))$,

and we assume that the leading coefficient matrix $D(x)$ of $\mathcal{D}_k$ belongs to $\mathcal{A}_n$. Any operator of the form (4.6) with arbitrary invertible matrix $D(x) \in \mathbb{K}[[x]]^{n\times n}$ is equivalent, by means of multiplications by unimodular matrices of $\mathbb{K}[[x]]^{n\times n}$, to an operator having a leading coefficient matrix in $\mathcal{A}_n$. Indeed, from Lemma 3.5.1 of Chapter 3, we know that any invertible matrix $D(x) \in \mathbb{K}[[x]]^{n\times n}$ can be written as

$$D(x) = P(x)\, x^{\alpha}\, Q(x),$$

where $P(x) \in \mathbb{K}[x]^{n\times n}$ with $\det(P(x)) = 1$, $Q(x) \in \mathbb{K}[[x]]^{n\times n}$ with $\det(Q(0)) \neq 0$ and $\alpha = \operatorname{diag}(\alpha_1 I_{n_1}, \ldots, \alpha_s I_{n_s})$ where the α_i's are nonnegative integers satisfying $\alpha_1 < \cdots < \alpha_s$. It follows that

$$P^{-1}(x)\, D(x)\, Q^{-1}(0) = x^{\alpha}\, Q(x)\, Q^{-1}(0) \in \mathcal{A}_n.$$

Hence, for any matrix differential operator $\mathcal{D}_k$ of the form (4.6) with arbitrary invertible matrix $D(x) \in \mathbb{K}[[x]]^{n\times n}$, there exist a unimodular matrix polynomial $S(x)$ and an invertible constant matrix T such that $S(x)\,\mathcal{D}_k\, T$ has a leading coefficient matrix in $\mathcal{A}_n$. Note that starting from a system of the form (4.1), it is easy to write it in the form (4.6) with $D(x) \in \mathcal{A}_n$; one can proceed as in the proof of Lemma 4.2.1 and obtain an operator of the form (4.6) with a diagonal leading coefficient matrix $D(x)$. In this case, the resulting operator is not necessarily k-simple unless System (4.1) is $(p-k)$-reduced.

Definition 4.3.1. *Let $\mathcal{D}_k$ be a matrix differential operator of the form (4.6) with $D(x) \in \mathcal{A}_n$. We define the* multi-index $\alpha(\mathcal{D}_k)$ *by $\alpha(\mathcal{D}_k) = (v(D(1,1)), \ldots, v(D(n,n)))$. In the sequel, we mean by $\alpha_i(\mathcal{D}_k)$ the ith component of $\alpha(\mathcal{D}_k)$.*

The main result of this section can be stated as follows.

Theorem 4.3.1. *Given a non-simple matrix differential operator $\mathcal{D}_k$ of the form (4.6) with a leading coefficient matrix $D(x) \in \mathcal{A}_n$, there exist two invertible matrices $U(x) \in \mathbb{K}[x^{-1}]^{n\times n}$ with $U^{-1}(x) \in \mathbb{K}[x]^{n\times n}$ and $V(x) \in \mathbb{K}[x]^{n\times n}$ with $V^{-1}(x) \in \mathbb{K}[x^{-1}]^{n\times n}$ such that the operator $U(x)\,\mathcal{D}_k\, V(x)$ is k-simple and satisfies $|\alpha(U(x)\,\mathcal{D}_k\, V(x))| < |\alpha(\mathcal{D}_k)|$.*

Assuming that the leading coefficient matrix $D(x)$ of $\mathcal{D}_k$ is in $\mathcal{A}_n$, the matrix pencil $L_k(\lambda)$ associated with $\mathcal{D}_k$ is then of the form

$$L_k(\lambda) = D_0\,\lambda + N_0 = \begin{pmatrix} I_r\,\lambda + N_0^{11} & N_0^{12} \\ N_0^{21} & N_0^{22} \end{pmatrix}, \tag{4.7}$$

where $r = \operatorname{rank}(D_0) < n$ (if $\operatorname{rank}(D_0) = n$, then $L_k(\lambda)$ is regular and hence $\mathcal{D}_k$ is k-simple) and N_0^{11} and N_0^{22} are square matrices of size respectively r and $n-r$. Note that all the rows of the submatrix $\begin{pmatrix} N_0^{21} & N_0^{22} \end{pmatrix}$ are nonzero since the matrices $D(x)$ and $N(x)$ are supposed to satisfy $\min\{v(D(i,.)), v(N(i,.))\} = 0$ for $i = 1, \ldots, n$. In the sequel, for a matrix pencil $L_k(\lambda)$ of the form (4.7), we will refer to the rows of $\begin{pmatrix} N_0^{21} & N_0^{22} \end{pmatrix}$ as the *constant rows* of $L_k(\lambda)$.

To prove Theorem 4.3.1, we first investigate the case where the constant rows of the matrix pencil $L_k(\lambda)$ are linearly dependent. In this case, we show that by multiplying $\mathcal{D}_k$ from two sides by suitable invertible matrices, we can drop the value of $|\alpha(\mathcal{D}_k)|$. Then, our purpose would be to assure that any operator $\mathcal{D}_k$ can be reduced, without increasing $|\alpha(\mathcal{D}_k)|$, to another matrix differential operator whose associated matrix pencil has linearly dependent constant rows. Note that the transformations carried out on the operator in all different stages may affect the special structure of the leading coefficient matrix and hence that of the associated matrix pencil. Each time this happens, we do additional transformations in order to ensure that the leading coefficient matrix still belongs to $\mathcal{A}_n$.

4.3.1 Linearly dependent constant rows

A sufficient (but not necessary) condition for the operator $\mathcal{D}_k$ to be non-simple is that the constant rows of its associated matrix pencil $L_k(\lambda)$ are linearly dependent. In this subsection, we show how to use these linear dependencies to reduce the value of $|\alpha(\mathcal{D}_k)|$.

Proposition 4.3.1. *Consider a matrix differential operator $\mathcal{D}_k$ of the form (4.6) with a leading coefficient matrix $D(x)$ in $\mathcal{A}_n$ and with $\min\{v(D(i,.)), v(N(i,.))\} = 0$ for $i = 1, \ldots, n$. Let $L_k(\lambda)$ denote the matrix pencil associated with $\mathcal{D}_k$. Assume that $r = \operatorname{rank}(D(0)) < n$ and the row of $L_k(\lambda)$ of index $i_0 \in \{r+1, \ldots, n-1\}$ can be written as a linear combination of the rows of index $i_0 + 1$ to n. Then, there exist an invertible constant matrix T and an invertible matrix $S(x) \in \mathbb{K}[x^{-1}]^{n\times n}$ of the form*

$$S(x) = P \operatorname{diag}(1, \ldots, 1, x^{-\gamma}, 1, \ldots, 1)\, S_1,$$

where P is a permutation matrix of size n, $S_1 \in \mathrm{GL}_n(\mathbb{K})$, $\gamma \in \mathbb{N}^*$ and $x^{-\gamma}$ stands at the i_0th position, such that the operator $\widetilde{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,T$ has coefficients in $\mathbb{K}[[x]]$ and satisfies $\left|\alpha(\widetilde{\mathcal{D}}_k)\right| = |\alpha(\mathcal{D}_k)| - \gamma < |\alpha(\mathcal{D}_k)|$.

Proof. Write

$$L_k(i_0, .) = -u_{i_0+1}\, L_k(i_0+1, .) - \cdots - u_n\, L_k(n, .),$$

with $u_i \in \mathbb{K}$ for $i = i_0 + 1, \ldots, n$. Define the matrix S_1 obtained from the identity matrix of size n by replacing its i_0th row by the row vector $\begin{pmatrix} 0 & \cdots & 0 & 1 & u_{i_0+1} & \cdots & u_n \end{pmatrix}$, where 1 comes at the i_0th position, that is,

$$S_1 = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & u_{i_0+1} & \cdots & u_n \\ & & & & 1 & & \\ & (0) & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}. \tag{4.8}$$

Multiply operator $\mathcal{D}_k$ on the left by S_1. Let $\overline{\mathcal{D}}_k = \overline{D}(x)\,\vartheta_k + \overline{N}(x)$ denote the resulting operator. Since S_1 is a constant matrix, the matrix pencil associated with $\overline{\mathcal{D}}_k$ is equal to $S_1\, L_k(\lambda)$ and hence operator $\overline{\mathcal{D}}_k$ is also non-simple. Moreover, by definition of S_1, the i_0th row of $\overline{N}(0)$ is zero. We point out here that the leading coefficient matrix $\overline{D}(x)$ of $\overline{\mathcal{D}}_k$ may not be in $\mathcal{A}_n$. Indeed, we have

$$v(\overline{D}(i_0, j)) = v(\overline{D}(i_0, i_0)) \quad \text{if } j > i_0 \text{ and } \alpha_j(\mathcal{D}_k) = \alpha_{i_0}(\mathcal{D}_k),$$
$$\text{and} \quad v(\overline{D}(i_0, j)) > v(\overline{D}(i_0, i_0)), \quad \text{otherwise.}$$

However, by multiplying operator $\overline{\mathcal{D}}_k$ on the right by

$$\overline{T} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & -\delta_{\alpha_{i_0},\alpha_{i_0+1}} u_{i_0+1} & \cdots & -\delta_{\alpha_{i_0},\alpha_n} u_n \\ & & & & 1 & & \\ & (0) & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}, \tag{4.9}$$

where for $j = i_0, \ldots, n$, α_j denotes $\alpha_j(\mathcal{D}_k)$, we get the operator $\widehat{\mathcal{D}}_k = \overline{\mathcal{D}}_k \overline{T}$ whose leading coefficient matrix belongs to $\mathcal{A}_n$. Put $\widehat{\mathcal{D}}_k = \widehat{D}(x)\,\vartheta_k + \widehat{N}(x)$. We have $\alpha(\widehat{\mathcal{D}}_k) = \alpha(\mathcal{D}_k)$ and the i_0th row of $\widehat{N}(x)$ is of valuation greater than or equal to 1 since the i_0th row of $\overline{N}(x)$ is so. As $\alpha_{i_0}(\widehat{\mathcal{D}}_k) = \alpha_{i_0}(\mathcal{D}_k)$ is a positive integer, a simplification of the i_0th row of $\widehat{\mathcal{D}}_k$ by a positive power of x is possible. So let $\gamma = \min(\alpha_{i_0}(\widehat{\mathcal{D}}_k), v(\widehat{N}(i_0, .))) \in \mathbb{N}^*$ and define

$$S_2(x) = \mathrm{diag}\,(1, \ldots, 1, x^{-\gamma}, 1, \ldots, 1), \tag{4.10}$$

where $x^{-\gamma}$ stands at the i_0th position. The multiplication of $\widehat{\mathcal{D}}_k$ on the left by $S_2(x)$ followed, if necessary, by a permutation of rows and columns[2] yields an operator $\widetilde{\mathcal{D}}_k \in \mathbb{K}[[x]][\vartheta_k]^{n\times n}$ with a leading coefficient matrix in $\mathcal{A}_n$ such that $\left|\alpha(\widetilde{\mathcal{D}}_k)\right| = |\alpha(\mathcal{D}_k)| - \gamma$. Finally, the matrix $S(x)$ we are seeking for is the product of a permutation matrix, $S_2(x)$ and S_1, and T is equal to $\overline{T}$ given by (4.9) up to a permutation of its columns. □

Remark 4.3.1. At the end of the proof of Proposition 4.3.1, we either have $\gamma = \alpha_{i_0}(\widehat{\mathcal{D}}_k)$ or $\gamma < \alpha_{i_0}(\widehat{\mathcal{D}}_k)$. If the former case occurs then, after simplifying the i_0th row of operator $\widehat{\mathcal{D}}_k$ by $x^{-\gamma}$, the i_0th diagonal entry of the leading coefficient matrix becomes of valuation zero. This means that the rank of the leading coefficient matrix of the associated matrix pencil has been increased by 1. Otherwise, the i_0th row of matrix $S_2(x)\,\widehat{N}(x)$ is of valuation zero and hence a new constant row has been introduced in the associated matrix pencil.

We now illustrate Proposition 4.3.1 with the following example.

Example 4.3.1. *Consider the matrix differential operator given by*

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} -2\,x^2 & 1 & -1 & 0 \\ 0 & -x^3 & 0 & -1 \\ 0 & 2\,x & -x & 1 \\ 0 & 0 & 0 & -x^3 - 1 \end{pmatrix}. \tag{4.11}$$

Here the multi-index $\alpha(\mathcal{D})$ is given by $\alpha(\mathcal{D}) = (0, 1, 1, 2)$ and $|\alpha(\mathcal{D})| = 4$. The matrix pencil associated with $\mathcal{D}$

$$L(\lambda) = \begin{pmatrix} \lambda & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

is singular since we have $L(2, .) = -L(3, .)$. Multiply operator $\mathcal{D}$ on the left by the invertible constant matrix

$$S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

to eliminate the linear dependency between the second and third row of $L(\lambda)$. We obtain the operator

$$S_1\,\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} -2\,x^2 & 1 & -1 & 0 \\ 0 & 2\,x - x^3 & -x & 0 \\ 0 & 2\,x & -x & 1 \\ 0 & 0 & 0 & -x^3 - 1 \end{pmatrix}$$

[2] To make the leading coefficient matrix in $\mathcal{A}_n$, more precisely, to make the valuations of diagonal entries in an increasing order.

whose leading coefficient matrix does not belong to $\mathcal{A}_4$. *For this reason, we multiply* $S_1\,\mathcal{D}$ *on the right by the matrix*

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.12}$$

We find the operator

$$\widehat{\mathcal{D}} = S\,\mathcal{D}\,T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} -2\,x^2 & 1 & -2 & 0 \\ 0 & 2\,x - x^3 & x^3 - 3\,x & 0 \\ 0 & 2\,x & -3\,x & 1 \\ 0 & 0 & 0 & -x^3 - 1 \end{pmatrix}$$

whose second row can be simplified by x. *Once this simplification is made,* i.e., *after multiplying* $\widehat{\mathcal{D}}$ *on the left by* $S_2(x) = \operatorname{diag}(1, x^{-1}, 1, 1)$, *we get the operator*

$$\widetilde{\mathcal{D}} = S_2(x)\,\widehat{\mathcal{D}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} -2\,x^2 & 1 & -2 & 0 \\ 0 & 2 - x^2 & x^2 - 3 & 0 \\ 0 & 2\,x & -3\,x & 1 \\ 0 & 0 & 0 & -x^3 - 1 \end{pmatrix} \tag{4.13}$$

which satisfies $|\alpha(\widetilde{\mathcal{D}})| = |(0,0,1,2)| = 3 = |\alpha(\mathcal{D})| - 1$. *Remark here that the rank of the leading coefficient matrix of the associated matrix pencil has been increased by* 1. ■

Lemma 4.3.1. *Under the assumptions of Proposition 4.3.1, let* ν *be an integer greater than or equal to* $|\alpha(\mathcal{D}_k)|$ *and let* $\mathcal{D}_{k_{|\nu}} = D(x)_{|\nu}\,\vartheta_k + N(x)_{|\nu}$ *denote the operator* $\mathcal{D}_k$ *truncated at order* ν. *Computing matrices* $S(x)$ *and* T *as in Proposition 4.3.1 and the operator* $S(x)\,\mathcal{D}_{k_{|\nu}}\,T$ *can be done using* $O\big(n^3 + \nu\,n^2\big)$ *arithmetic operations in* $\mathbb{K}$.

Proof. First, we need to compute a vector in the left nullspace of an $(n-r)\times n$ constant matrix. Using Gaussian elimination, this can be done in $O\big(n^3\big)$ operations in $\mathbb{K}$. Then, we compute the products $S_1\,D(x)_{|\nu}$ and $S_1\,N(x)_{|\nu}$, where S_1 is the matrix given by (4.8). Due to the particular structure of S_1, this costs $O\big(n^2\,\nu\big)$ operations in $\mathbb{K}$. Similarly, the multiplication on the right by matrix $\overline{T}$ given by (4.9) costs $O\big(n^2\,\nu\big)$ operations in $\mathbb{K}$. Finally, as the multiplication on the left by matrix $S_2(x)$ given by (4.10) is just a simplification by a positive power of x, computing matrices $S(x)$ and T and the operator $S(x)\,\mathcal{D}_{k_{|\nu}}\,T$ can be done in $O\big(n^3 + n^2\,\nu\big)$ operations in $\mathbb{K}$. □

After applying the process described in the proof of Proposition 4.3.1 to $\mathcal{D}_k$, if the constant rows of the matrix pencil associated with operator $\widetilde{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,T$ are yet linearly dependent, then we repeat the same process but now on $\widetilde{\mathcal{D}}_k$. Thus, as long as the constant rows of the matrix pencil associated with the resulting operator are linearly dependent, by applying Proposition 4.3.1, we can decrease the value of $|\alpha|$ by at least 1. Consequently, after at most $|\alpha(\mathcal{D}_k)|$ successive iterations, we get an operator equivalent to $\mathcal{D}_k$ satisfying one of the three following cases:

- the length of its multi-index α is zero, or, equivalently, the leading coefficient matrix of its associated matrix pencil is the identity matrix I_n in which case the operator is k-simple and we are done;
- the length of its multi-index α is nonzero, the constant rows of its associated matrix pencil are linearly independent and the operator is k-simple;

- the length of its multi-index α is nonzero, the constant rows of its associated matrix pencil are linearly independent but the operator is still non simple. We will see in the next subsection how this operator can be transformed, without affecting the multi-index α, into another one whose associated matrix pencil has linearly dependent constant rows. Hence, the length of α can be further reduced (by going back to Proposition 4.3.1).

Example 4.3.2. *We continue our reduction process on operator $\widetilde{\mathcal{D}}$ given in (4.13) which is yet non-simple since the constant rows of its associated matrix pencil are still linearly dependent. Proposition 4.3.1 applied to $\widetilde{\mathcal{D}}$ gives rise to the operator*

$$\overline{\mathcal{D}} = \overline{D}(x)\,\vartheta + \overline{N}(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} -2\,x^2 & 1 & -2 & 0 \\ 0 & -x^2+2 & x^2-3 & 0 \\ 0 & 2 & -3 & -x^2 \\ 0 & 0 & 0 & -x^3-1 \end{pmatrix} \tag{4.14}$$

which is simple and satisfies $|\alpha(\overline{\mathcal{D}})| = 2 = |\alpha(\widetilde{\mathcal{D}})| - 1$. Finally, the two operators $\overline{\mathcal{D}}$ and $\mathcal{D}$ given in (4.11) are related by $\overline{\mathcal{D}} = \overline{S}(x)\,\mathcal{D}\,T$, where

$$\overline{S}(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x^{-1} & x^{-1} & 0 \\ 0 & 0 & x^{-1} & x^{-1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and T is given by (4.12). ■

Remark 4.3.2. Consider the linear differential system $\overline{\mathcal{D}}(y(x)) = 0$, where $\overline{\mathcal{D}}$ is defined in (4.14). Writing System $\overline{\mathcal{D}}(y(x)) = 0$ in the form $\vartheta(y(x)) = A(x)\,y(x)$ with

$$A(x) = \overline{D}^{-1}(x)\,\overline{N}(x) = \begin{pmatrix} 2\,x^2 & -1 & 2 & 0 \\ 0 & x^2-2 & 3-x^2 & 0 \\ 0 & -2 & 3 & -\frac{1}{x} \\ 0 & 0 & 0 & x+\frac{1}{x^2} \end{pmatrix},$$

the function $\theta_2(A,x,\lambda)$ defined in (4.5) is equal to

$$\theta_2(A,x,\lambda) = x\left(\lambda - 2\,x^2\right)\left(\lambda\,x^2 - x^3 - 1\right)\left(\lambda^2 - \lambda - \lambda\,x^2 + x^2\right)$$

and hence $\theta_2(A,0,\lambda) = 0$. It follows from Proposition 4.2.1 that the system is not super-reduced.

4.3.2 Reduction to the case of linearly dependent constant rows

Consider a non-simple matrix differential operator $\mathcal{D}_k$ of the form (4.6) with a leading coefficient matrix $D(x) \in \mathcal{A}_n$. We show in this subsection how operator $\mathcal{D}_k$ can be reduced, without affecting the multi-index $\alpha(\mathcal{D}_k)$, to another operator to which one can apply Proposition 4.3.1, that is, the associated matrix pencil of which has linearly dependent constant rows. We proceed in a similar way as in [58]. At first, we investigate the case where the matrix pencil associated with operator $\mathcal{D}_k$ has the particular block structure

$$L_k(\lambda) = \begin{pmatrix} I_q\,\lambda + N_0^{11} & 0 & 0 \\ N_0^{21} & I_{r-q}\,\lambda + N_0^{22} & N_0^{23} \\ N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix}, \tag{4.15}$$

where $q \in \{0,\ldots,r\}$, N_0^{11}, N_0^{22} and N_0^{33} are square matrices respectively of size q, $r-q$ and $n-r$, and

$$\operatorname{rank}\begin{pmatrix} N_0^{32} & N_0^{33} \end{pmatrix} < n - r. \tag{4.16}$$

We show that by multiplying $\mathcal{D}_k$ from both sides by appropriate invertible matrices, we get an operator whose associated matrix pencil has linearly dependent constant rows. Secondly, we explain how we can transform operator $\mathcal{D}_k$ into another one having an associated matrix pencil of the form (4.15) with condition (4.16) satisfied.

Proposition 4.3.2. *Consider a non-simple matrix differential operator $\mathcal{D}_k = D(x)\,\vartheta_k + N(x) \in \mathbb{K}[[x]][\vartheta_k]^{n\times n}$ with $D(x) \in \mathcal{A}_n$ and let r denote $\operatorname{rank}(D(0))$. Assume that the matrix pencil $L_k(\lambda)$ associated with $\mathcal{D}_k$ is of the form (4.15) with condition (4.16) satisfied. By multiplying $\mathcal{D}_k$ on the left by*

$$S(x) = \operatorname{diag}(x^{-1}\,I_q, I_{r-q}, I_{n-r}), \tag{4.17}$$

then on the right by

$$T(x) = S^{-1}(x)\,C, \tag{4.18}$$

where C is a suitable invertible constant matrix of the form

$$C = \begin{pmatrix} I_q & A & B \\ 0 & I_{r-q} & 0 \\ 0 & 0 & I_{n-r} \end{pmatrix},$$

one gets an operator $\widetilde{\mathcal{D}}_k$ satisfying $\alpha(\widetilde{\mathcal{D}}_k) = \alpha(\mathcal{D}_k)$ and whose associated matrix pencil has linearly dependent constant rows.

Proof. If $q = 0$ then the constant rows of the matrix pencil $L_k(\lambda)$ given in (4.15) are already linearly dependent and hence nothing has to be done. Otherwise, partition matrices $D(x)$ and $N(x)$ into blocks of the same sizes as those of $L_k(\lambda)$ given in (4.15), *i.e.*,

$$D(x) = \begin{pmatrix} D^{11} & D^{12} & D^{13} \\ D^{21} & D^{22} & D^{23} \\ D^{31} & D^{32} & D^{33} \end{pmatrix} \quad \text{and} \quad N(x) = \begin{pmatrix} N^{11} & N^{12} & N^{13} \\ N^{21} & N^{22} & N^{23} \\ N^{31} & N^{32} & N^{33} \end{pmatrix},$$

where D^{11}, D^{22} and D^{33}, respectively N^{11}, N^{22} and N^{33}, are square matrices of respective size q, $r-q$ and $n-r$ (the dependence on x has been omitted in the notation of the blocks D^{ij} and N^{ij} for brevity). Set $\overline{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,S^{-1}(x) = \overline{D}(x)\,\vartheta_k + \overline{N}(x)$, where $S(x)$ is the matrix defined in (4.17). We have

$$\overline{D}(x) = \begin{pmatrix} D^{11} & x^{-1}D^{12} & x^{-1}D^{13} \\ x\,D^{21} & D^{22} & D^{23} \\ x\,D^{31} & D^{32} & D^{33} \end{pmatrix}$$

and

$$\begin{aligned} \overline{N}(x) &= S(x)\,N(x)\,S^{-1}(x) + S(x)D(x)\,\vartheta_k\big(S^{-1}(x)\big) \\ &= \begin{pmatrix} N^{11} + x^k\,D^{11} & x^{-1}N^{12} & x^{-1}N^{13} \\ x\,N^{21} + x^{k+1}D^{21} & N^{22} & N^{23} \\ x\,N^{31} + x^{k+1}D^{31} & N^{32} & N^{33} \end{pmatrix}. \end{aligned}$$

Since $L_k(\lambda)$ is of the form (4.15), the blocks D^{12}, D^{13}, N^{12} and N^{13} are of positive valuations and hence the matrices $\overline{D}(x)$ and $\overline{N}(x)$ have entries in $\mathbb{K}[[x]]$. Set $\overline{D}^{1j} = x^{-1}D^{1j}$ and $\overline{N}^{1j} = x^{-1}N^{1j}$ for $j = 2, 3$. The matrix pencil $\overline{L}_k(\lambda)$ associated with operator $\overline{\mathcal{D}}_k$ is then equal to

$$\overline{L}_k(\lambda) = \begin{pmatrix} I_q\,\lambda + N_0^{11} + \delta_{k,0}\,I_q & \overline{D}_0^{12}\lambda + \overline{N}_0^{12} & \overline{D}_0^{13}\lambda + \overline{N}_0^{13} \\ 0 & I_{r-q}\,\lambda + N_0^{22} & N_0^{23} \\ 0 & N_0^{32} & N_0^{33} \end{pmatrix}.$$

To bring it back to the form (4.7), we will multiply operator $\overline{\mathcal{D}}_k$ on the right by the constant invertible matrix

$$C = \begin{pmatrix} I_q & -\overline{D}_0^{12} & -\overline{D}_0^{13} \\ 0 & I_{r-q} & 0 \\ 0 & 0 & I_{n-r} \end{pmatrix}. \tag{4.19}$$

This gives rise to an operator $\widetilde{\mathcal{D}}_k = \widetilde{D}(x)\,\vartheta_k + \widetilde{N}(x)$ whose leading coefficient matrix $\widetilde{D}(x)$ belongs to $\mathcal{A}_n$. Moreover, the diagonal entries of $\widetilde{D}(x)$ and $D(x)$ have the same valuations which implies that the multi-index $\alpha(\widetilde{\mathcal{D}}_k)$ is equal to $\alpha(\mathcal{D}_k)$. Now, the matrix pencil $\widetilde{L}_k(\lambda)$ associated with $\widetilde{\mathcal{D}}_k$ is given by

$$\widetilde{L}_k(\lambda) = \overline{L}_k(\lambda)\,C = \begin{pmatrix} I_q\,\lambda + N_0^{11} + \delta_{k,0}\,I_q & \overline{N}_0^{12} - N_0^{11}\overline{D}_0^{12} - \delta_{k,0}\overline{D}_0^{12} & \overline{N}_0^{13} - N_0^{11}\overline{D}_0^{13} - \delta_{k,0}\overline{D}_0^{13} \\ 0 & I_{r-q}\,\lambda + N_0^{22} & N_0^{21} \\ 0 & N_0^{32} & N_0^{33} \end{pmatrix}$$

and has linearly dependent constant rows from (4.16). Thus, the proposition is proved. □

Lemma 4.3.2. *Under the assumptions of Proposition 4.3.2, let ν be an integer greater than or equal to $|\alpha(\mathcal{D}_k)|$ and let $\mathcal{D}_{k_{|\nu}} = D(x)_{|\nu}\,\vartheta_k + N(x)_{|\nu}$ denote the operator $\mathcal{D}_k$ truncated at order ν. Computing matrices $S(x)$ and $T(x)$ as in Proposition 4.3.2 and the operator $S(x)\,\mathcal{D}_{k_{|\nu}}\,T(x)$ can be done using $O(\nu\,n^3)$ operations in $\mathbb{K}$.*

Proof. We have just to study the cost of the product $\overline{\mathcal{D}}_{k_{|\nu}}\,C$, where C is given by (4.19) and $\overline{\mathcal{D}}_k = S(x)\,\mathcal{D}_{k_{|\nu}}\,S^{-1}(x)$ with $S(x)$ given by (4.17). This needs at most $2\,n\,q\,(n-q)\,\nu$ multiplications in $\mathbb{K}$. As $q \leq r < n$, we find $O(\nu\,n^3)$ operations in $\mathbb{K}$. □

Now it remains to explain how any non-simple matrix differential operator can be reduced to another operator with same multi-index α and whose associated matrix pencil is of the form (4.15), with condition (4.16) satisfied. This can be done using only constant transformations as shown in the following proposition.

Proposition 4.3.3. *Let $\mathcal{D}_k = D(x)\,\vartheta_k + N(x) \in \mathbb{K}[[x]][\vartheta_k]^{n\times n}$ be a non-simple matrix differential operator having a leading coefficient matrix $D(x)$ in $\mathcal{A}_n$. There exist two invertible constant matrices $S, T \in \mathrm{GL}_n(\mathbb{K})$ such that the operator $S\,\mathcal{D}_k\,T$ satisfies $\alpha(S\,\mathcal{D}_k\,T) = \alpha(\mathcal{D}_k)$ and has an associated matrix pencil of the form (4.15), with (4.16) satisfied.*

Proof. If the constant rows of the matrix pencil $L_k(\lambda)$ associated with $\mathcal{D}_k$ are linearly dependent then $L_k(\lambda)$ is already of the form (4.15) with $q = 0$ and condition (4.16) satisfied. In this case, we have $S = T = I_n$. Suppose now that $L_k(\lambda)$ is given in (4.7) and that its constant rows are linearly independent. Since $\mathcal{D}_k$ is non-simple, $L_k(\lambda)$ is then singular for all $\lambda \in \mathbb{C}$. In particular, for $\lambda = 0$, the constant matrix

$$L(0) = N(0) = \begin{pmatrix} N_0^{11} & N_0^{12} \\ N_0^{21} & N_0^{22} \end{pmatrix}$$

is singular as well. Therefore, there exists a nonzero row vector $u = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix} \in \mathbb{K}^{1\times n}$ such that $u\,L(0) = 0$. As the constant rows of $L_k(\lambda)$ are supposed to be linearly independent, one component of u with index $i \in \{1, \ldots, r\}$ is necessarily nonzero. Swapping the rows and columns of $\mathcal{D}_k$ of index 1 to r, we can achieve $i = 1$. Furthermore, without any loss of generality, we will assume that $u_1 = 1$. Multiply operator $\mathcal{D}_k$ on the left by

$$S_1 = \begin{pmatrix} 1 & u_2 & u_3 & \cdots & u_n \\ & 1 & & & \\ & & 1 & & \\ & (0) & & \ddots & \\ & & & & 1 \end{pmatrix}$$

and let $\widehat{\mathcal{D}}_k = \widehat{D}(x)\,\vartheta_k + \widehat{N}(x)$ denote the resulting operator. It follows that the first row of $\widehat{N}(0)$ is equal to $u\,N(0) = u\,L(0) = 0$. Note here that the leading coefficient matrix $\widehat{D}(x)$ of $\widehat{\mathcal{D}}_k$ may not be in $\mathcal{A}_n$ since it could happen that for some $j \in \{2, \ldots, r\}$, we have $v(\widehat{D}(1, j)) = 0$[3]. Therefore, multiply $\widehat{\mathcal{D}}_k$ on the right by

$$T_1 = \begin{pmatrix} 1 & -u_2 & \cdots & -u_r & 0 & \cdots & 0 \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & (0) & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

and put $\widetilde{\mathcal{D}}_k = \widehat{\mathcal{D}}_k\,T_1 = \widetilde{D}(x)\,\vartheta_k + \widetilde{N}(x)$. We have $\widetilde{D}(x) \in \mathcal{A}_n$, $\alpha(\widetilde{\mathcal{D}}_k) = \alpha(\mathcal{D}_k)$ and the first row of $\widetilde{N}(0)$ is zero. The matrix pencil $\widetilde{L}_k(\lambda)$ associated with $\widetilde{\mathcal{D}}_k$ is then of the form

$$\widetilde{L}_k(\lambda) = \begin{pmatrix} \lambda & 0 & 0 \\ \widetilde{N}_0^{21} & I_{r-1}\,\lambda + \widetilde{N}_0^{22} & \widetilde{N}_0^{23} \\ \widetilde{N}_0^{31} & \widetilde{N}_0^{32} & \widetilde{N}_0^{33} \end{pmatrix}.$$

We draw attention that $\widetilde{\mathcal{D}}_k$ is also non-simple since $\widetilde{L}_k(\lambda) = S_1\,L_k(\lambda)\,T_1$. Now, if the rows of the submatrix $\begin{pmatrix} \widetilde{N}_0^{32} & \widetilde{N}_0^{33} \end{pmatrix}$ are linearly dependent then $\widetilde{L}_k(\lambda)$ has the form (4.15) with $q = 1$ and condition (4.16) holds. Otherwise, due to the structure of $\widetilde{L}_k(\lambda)$, we can deduce that the submatrix

$$\begin{pmatrix} I_{r-1}\,\lambda + \widetilde{N}_0^{22} & \widetilde{N}_0^{23} \\ \widetilde{N}_0^{32} & \widetilde{N}_0^{33} \end{pmatrix}$$

is singular. One can therefore continue the process and increment q by 1. Thus, after at most r successive iterations of the process, we obtain an equivalent non-simple operator whose associated matrix pencil is either of the form (4.15) with $q < r$ and condition (4.16) occurs, or $q = r$ which means that it is of the form

$$\begin{pmatrix} I_r\,\lambda + W_1 & 0 \\ W_2 & W_3 \end{pmatrix}. \tag{4.20}$$

As the matrix in (4.20) is singular and the first diagonal block $I_r\,\lambda + W_1$ is regular, the submatrix W_3 is necessarily singular. Consequently, the matrix in (4.20) has the form (4.15) with $q = r$ and condition (4.16) is satisfied. Finally, the matrices S and T we are seeking for are the product of permutation matrices and upper constant matrices. □

The procedure described in the proof above produces an operator having an associated matrix pencil in the form (4.15) with a strictly lower triangular matrix N_0^{11}. However, to apply Proposition 4.3.2, N_0^{11} needs not to be in this form.

Lemma 4.3.3. *Let $\mathcal{D}_k = D(x)\,\vartheta_k + N(x)$ be a non-simple matrix differential operator having a leading coefficient matrix $D(x)$ in $\mathcal{A}_n$. Let ν be an integer greater than or equal to $|\alpha(\mathcal{D}_k)|$ and let $\mathcal{D}_{k_{|\nu}} = D(x)_{|\nu}\,\vartheta_k + N(x)_{|\nu}$ denote the operator $\mathcal{D}_k$ truncated at order ν. Computing matrices S and T as in Proposition 4.3.3 and the operator $S\,\mathcal{D}_{k_{|\nu}}\,T$, whose associated matrix pencil is of the form (4.15) with (4.16) satisfied, can be done using $O(n^{\omega+1} + \nu\,n^3)$ arithmetic operations in $\mathbb{K}$.*

[3] This occurs when $u_j \neq 0$.

Proof. Computing an operator such that its associated matrix pencil is of the form (4.15) with (4.16) satisfied requires at most r steps where $r = \operatorname{rank}(D(0)) < n$. At each step i, we consider an $(n-i+1) \times (n-i+1)$ matrix pencil and compute a vector in the left nullspace of this matrix evaluated at $\lambda = 0$. This can be done using $O((n-i+1)^\omega)$ operations in $\mathbb{K}$. Therefore, the total cost is

$$\sum_{i=1}^{r} (n-i+1)^\omega \leq r\, n^\omega < n^{\omega+1}.$$

Let now $\mathcal{D}_k^0$ denote the operator $\mathcal{D}_{k_{|\nu}}$. At each step i, we form two constant matrices S_i and T_i, each of them is the product of elementary matrices one of them being a permutation matrix, and compute the operator $\mathcal{D}_k^i = S_i\, \mathcal{D}_k^{i-1}\, T_i$. This requires at most $2\,\nu\,(n-i)\,n$ multiplications in $\mathbb{K}$. Thus, the final operator $S\, \mathcal{D}_{k_{|\nu}}\, T$ is obtained after at most

$$\sum_{i=1}^{r} 2\,\nu\,(n-i)\,n = 2\,\nu\,n \sum_{i=1}^{r} (n-i) < 2\,\nu\,n^2 r < 2\,\nu\,n^3$$

operations in $\mathbb{K}$. Now it remains to compute the costs of constructing matrices S and T. Matrix S, respectively T, is the product of at most r matrices S_i, respectively r matrices T_i. Due to the particular structures of these matrices, computing matrix T requires at most $n \sum_{i=1}^{r-1} (r-i)$ multiplications in $\mathbb{K}$ which is bounded by n^3 multiplications whereas computing matrix S costs nothing. Consequently, the total cost of computing S, T and $S\, \mathcal{D}_{k_{|\nu}}\, T$ is $O(n^{\omega+1} + \nu\, n^3)$ operations in $\mathbb{K}$. □

4.3.3 An example

Consider the matrix differential operator given by

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x^3 \end{pmatrix} \vartheta + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1-x^2 & 0 & 0 & 1 & -x \\ x^3-x & 0 & 0 & 1-x^2 & 0 \end{pmatrix}. \tag{4.21}$$

One can easily check that the matrix pencil associated with $\mathcal{D}$

$$L(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 1 & 0 \\ 0 & \lambda & 0 & 0 & 1 \\ 0 & 1 & \lambda & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

is singular therefore $\mathcal{D}$ is not simple. Moreover, Proposition 4.3.1 cannot be applied to $\mathcal{D}$ since the constant rows of $L(\lambda)$ are linearly independent. So let us first reduce operator $\mathcal{D}$ to another one whose associated matrix pencil is of the form (4.15) with $1 \leq q \leq 3$ and for which condition (4.16) is satisfied. To achieve this, we will follow the steps of the proof of Proposition 4.3.3. The left nullspace of matrix $L(0)$ contains the row vector $u = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \end{pmatrix}$. So define the matrix S_1 by

$$S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Here, the matrix T_1 appearing in the proof of Proposition 4.3.3 is equal to I_5. Multiplying $\mathcal{D}$ on the left by S_1, we find

$$\widetilde{\mathcal{D}} = S_1\,\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & -x^3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x^3 \end{pmatrix} \vartheta + \begin{pmatrix} x - x^3 & 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 - x^2 & 0 & 0 & 1 & -x \\ x^3 - x & 0 & 0 & 1 - x^2 & 0 \end{pmatrix}$$

whose associated matrix pencil

$$\widetilde{L}(\lambda) = \left(\begin{array}{c|cc|cc} \lambda & 0 & 0 & 0 & 0 \\ \hline 0 & \lambda & 0 & 0 & 1 \\ 0 & 1 & \lambda & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

has the form (4.15) with $q = 1$ and condition (4.16) satisfied since

$$\operatorname{rank} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = 1 < n - r = 2.$$

We can now apply Proposition 4.3.2 to the operator $\widetilde{\mathcal{D}}$. Let

$$S(x) = \operatorname{diag}\left(x^{-1}, 1, 1, 1, 1\right)$$

and compute $\overline{\mathcal{D}} = S(x)\,\widetilde{\mathcal{D}}\,S^{-1}(x)$. We find

$$\overline{\mathcal{D}} = \begin{pmatrix} 1 & 0 & 0 & 0 & -x^2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x^3 \end{pmatrix} \vartheta + \begin{pmatrix} x - x^3 + 1 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ x - x^3 & 0 & 0 & 1 & -x \\ x^4 - x^2 & 0 & 0 & 1 - x^2 & 0 \end{pmatrix}.$$

Here, the matrix C defined in (4.19) is the identity matrix since the leading coefficient matrix of $\overline{\mathcal{D}}$ is already in $\mathcal{A}_5$. Now the constant rows of the matrix pencil associated with $\overline{\mathcal{D}}$ are linearly dependent hence Proposition 4.3.1 can be applied to $\overline{\mathcal{D}}$. This results in the operator

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -x^2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -x^2 \\ 0 & 0 & 0 & 0 & x^3 \end{pmatrix} \vartheta + \begin{pmatrix} x - x^3 + 1 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 - x^2 + x - x^3 & 0 & 0 & x & -1 \\ x^4 - x^2 & 0 & 0 & 1 - x^2 & 0 \end{pmatrix} \tag{4.22}$$

which is simple since the determinant of its associated matrix pencil is equal to $\lambda^2(\lambda + 1)$. The latter operator is equivalent to the operator $\mathcal{D}$ given by (4.21) by means of the two matrices

$$S_f = \begin{pmatrix} \frac{1}{x} & 0 & 0 & 0 & -\frac{1}{x} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{x} & -\frac{1}{x} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad T_f = \operatorname{diag}(x, 1, 1, 1, 1).$$

Finally, it is worth mentioning that the system defined by the operator given in (4.22), written in the form

$$\vartheta(y(x)) - \begin{pmatrix} -1 & 0 & 0 & -x^{-1} & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 \\ 1+x^2 & 0 & 0 & -x^{-1} & 1 \\ -\frac{-1+x^2}{x} & 0 & 0 & \frac{-1+x^2}{x^3} & 0 \end{pmatrix} y(x) = 0, \tag{4.23}$$

is not Moser-reduced and hence not super-reduced. Indeed, using Moser's algorithm [77], one can compute a system gauge-equivalent to (4.23) having a Poincaré-rank equal to 2 while the Poincaré-rank of System (4.23) is 3. Consequently, our approach computes 0-simple forms which are not necessarily super-reduced.

4.4 Algorithm and complexity estimation

Combining the results of Propositions 4.3.1, 4.3.2 and 4.3.3, we derive the following algorithm which has been implemented[4] in Maple.

Algorithm **SimpleForm**

Input: The coefficient matrices D and N of the operator $\mathcal{D}_k = D\,\vartheta_k + N$ given by (4.6) truncated at order $\nu \geq |\alpha(\mathcal{D}_k)|$.

Output: Four matrices $U, V, \widetilde{D}$ and $\widetilde{N}$ with $U \in \mathbb{K}[x^{-1}]^{n\times n}$ and $V \in \mathbb{K}[x]^{n\times n}$ both invertible, $\widetilde{D} = U D V \mod x^{\nu+1}$ and $\widetilde{N} = U(N V + D\,\vartheta_k(V)) \mod x^{\nu+1}$ such that the operator $\widetilde{D}\,\vartheta_k + \widetilde{N}$ is k-simple.

Initialization: Put $U = I_n$, $V = I_n$, $\widetilde{D} = D$, $\widetilde{N} = N$ and $\widetilde{L} = \widetilde{D}_0\,\lambda + \widetilde{N}_0$;

While $\widetilde{L}$ is singular **do**

1. Compute two constant matrices $S, T \in \mathrm{GL}_n(\mathbb{K})$ as in Proposition 4.3.3 such that the matrix pencil $S\,\widetilde{L}\,T$ is of the form (4.15) with (4.16) satisfied;
2. Let $\widetilde{D} = S\,\widetilde{D}\,T$ and $\widetilde{N} = S\,\widetilde{N}\,T$;
3. Let $U = S\,U$ and $V = V\,T$;
4. Compute two matrices $S \in \mathbb{K}[x^{-1}]^{n\times n}$ and $T \in \mathbb{K}[x]^{n\times n}$ as in Proposition 4.3.2;
5. Let $\widetilde{D} = S\,\widetilde{D}\,T$ and $\widetilde{N} = S\left(\widetilde{N}\,T + \widetilde{D}\,\vartheta_k(T)\right)$;
6. Let $U = S\,U$ and $V = V\,T$;
7. Compute two matrices $S \in \mathbb{K}[x^{-1}]^{n\times n}$ and $T \in \mathrm{GL}_n(\mathbb{K})$ as in Proposition 4.3.1;
8. Let $\widetilde{D} = S\,\widetilde{D}\,T$ and $\widetilde{N} = S\,\widetilde{N}\,T$;
9. Let $U = S\,U$ and $V = V\,T$;
10. Let $\widetilde{L} = \widetilde{D}_0\,\lambda + \widetilde{N}_0$;

end do;

Return U, V, $\widetilde{D}_{|_\nu}$ and $\widetilde{N}_{|_\nu}$;

Proposition 4.4.1. *Algorithm* **SimpleForm** *stops after at most* $|\alpha(\mathcal{D}_k)|$ *calls of the* **While** *loop, where* $\mathcal{D}_k$ *denotes the operator given in input.*

Proof. In each passage through the **While** loop, the value of $|\alpha(\widetilde{D}\,\vartheta_k + \widetilde{N})|$ decreases at least by 1. This only happens at Step 8 (see Proposition 4.3.1). Consequently, the algorithm stops either

[4] The code is available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`

before carrying out $|\alpha(\mathcal{D}_k)|$ calls of the **While** loop or after the $\alpha(\mathcal{D}_k)$th call where we have $|\alpha(\widetilde{D}\,\vartheta_k + \widetilde{N})| = 0$. This means that $\widetilde{D}_0 = I_n$ and hence $\widetilde{D}\,\vartheta_k + \widetilde{N}$ is necessarily k-simple. □

In order to study the arithmetic complexity of the algorithm above, we give first a lower bound on the valuation of the output U.

Lemma 4.4.1. *The valuation of the output U of Algorithm **SimpleForm** is greater than or equal to $-|\alpha(\mathcal{D}_k)|$, where $\mathcal{D}_k$ denotes the operator given in the input.*

Proof. Let S_1, S_2 and S_3 denote the matrix S computed respectively in Steps 1, 4 and 7. In one passage through the **While** loop, matrices S_1 and S_2 act only on the first r ($r = \operatorname{rank}(\widetilde{D}_0)$) rows of the operator $\widetilde{D}\,\vartheta_k + \widetilde{N}$ whereas the matrix S_3 acts only on its last $n - r$ rows. Moreover, since $v(S_3) \leq v(S_2)$, the valuation of the product $S_3\,S_2\,S_1$ is then equal to that of S_3. Thus, the valuation of the output matrix U is greater than or equal to the sum of the valuations of the matrices S_3 constructed during the algorithm. Let $\widetilde{\mathcal{D}}_k$ denote the operator $\widetilde{D}\,\vartheta_k + \widetilde{N}$ just before Step 8 of the algorithm. According to Proposition 4.3.1, matrix S_3 satisfies $v(S_3) = -\left|\alpha(\widetilde{\mathcal{D}}_k)\right| + \left|\alpha(S_3\,\widetilde{\mathcal{D}}_k\,T)\right|$, where T is the matrix computed in Step 7. Since the multi-index α is unchanged in Steps 1 to 7 of the algorithm, the output matrix U satisfies then $v(U) \geq -|\alpha(\mathcal{D}_k)| + |\alpha(U\,\mathcal{D}_k\,V)|$. Now, we have $|\alpha(U\,\mathcal{D}_k\,V)| \geq 0$, so we find $v(U) \geq -|\alpha(\mathcal{D}_k)|$. □

As for the output matrix V, we can say that $\deg(V) \leq |\alpha(\mathcal{D}_k)|$, where $\mathcal{D}_k$ denotes the operator given in input.

Proposition 4.4.2. *Algorithm* **SimpleForm** *uses at most $O\big(n^{\omega+1}\,|\alpha(\mathcal{D}_k)| + \nu\,n^3\,|\alpha(\mathcal{D}_k)|\big)$ operations in $\mathbb{K}$.*

Proof. According to Lemma 4.3.3, Steps 1 and 2 can be done in at most $O\big(n^{\omega+1} + \nu\,n^3\big)$ operations in $\mathbb{K}$. From Lemma 4.3.2, Steps 4 and 5 cost at most $O\big(\nu\,n^3\big)$ operations in $\mathbb{K}$. From Lemma 4.3.1, Steps 7 and 8 cost at most $O\big(n^3 + \nu\,n^2\big)$ operations in $\mathbb{K}$. It remains to study the cost of Steps 3, 6 and 9. Due to the particular structure of matrices S and T and to the fact that the valuation of matrix U is always bounded by $-|\alpha(\mathcal{D}_k)|$ and the degree of matrix V is always bounded by $|\alpha(\mathcal{D}_k)|$, the cost of Steps 3, 6 and 9 is bounded by $O\big(n^3\,|\alpha(\mathcal{D}_k)|\big)$ operations in $\mathbb{K}$. Since $\nu \geq |\alpha(\mathcal{D}_k)|$, one passage in the **While** loop costs then $O\big(n^{\omega+1} + \nu\,n^3\big)$ operations in $\mathbb{K}$. As it is repeated at most $|\alpha(\mathcal{D}_k)|$ times (see Proposition 4.4.1), Algorithm **SimpleForm** returns a k-simple operator using $O\big(n^{\omega+1}\,|\alpha(\mathcal{D}_k)| + \nu\,n^3\,|\alpha(\mathcal{D}_k)|\big)$ operations in $\mathbb{K}$. □

Remark 4.4.1. If $\mathcal{A}_n$ denotes the set of invertible matrices M of $\mathbb{K}[[x]]^{n\times n}$ which satisfy in addition to properties 1 to 3 on page 98 the following property

4. $\forall\, i = 1, \ldots, n$ and $j = 1, \ldots, n$ such that $i \neq j$, $v(M(i,j)) > v(M(j,j))$,

then an analogous result to that in Proposition 4.3.1 can be stated on the constant columns of the matrix pencil $L_k(\lambda)$ given by (4.7), *i.e.*, on the columns of the submatrix $\begin{pmatrix} N_0^{12} \\ N_0^{22} \end{pmatrix}$. Hence, one has a variant of Algorithm **SimpleForm** which consists first in eliminating all the linear dependencies between the constant rows and between the constant columns of the associated matrix pencil then applying Propositions 4.3.3 and 4.3.2 and so on. This variant could be more efficient in practice but its complexity estimation is the same as Algorithm **SimpleForm** above.

Remark 4.4.2. Given a linear differential system $\mathcal{D}(y(x)) = D(x)\,\vartheta(y(x)) + N(x)\,y(x) = 0$ with $D(x) \in \mathcal{A}_n$ and $N(x) \in \mathbb{K}[[x]]^{n\times n}$, Algorithm **SimpleForm** applied to $\mathcal{D}$ allows to detect whether the point $x = 0$ is a regular or an irregular singularity of the system $\mathcal{D}(y(x)) = 0$. Indeed, let $\widetilde{L}(\lambda)$ denote the matrix pencil associated with the simple operator $\widetilde{D}_{|\nu}\,\vartheta + \widetilde{N}_{|\nu}$

produced by the algorithm. Hence, System $\mathcal{D}(y(x)) = 0$ has a regular singularity at $x = 0$ if and only if $\deg(\det(\widetilde{L}(\lambda))) = n$. This occurs only if $\left|\alpha(\widetilde{D}_{|\nu}\,\vartheta + \widetilde{N}_{|\nu})\right| = 0$.

Remark 4.4.3. Given a linear differential system of the form (4.1), it is well known that the polynomials q_i in (4.2) are determined by the coefficients $A_0, A_1, \ldots, A_{np-1}$ of the system matrix $A(x)$, see [7, 13]. So let $\nu \geq np$ denote the number of coefficients in the expansion of $A(x)$ taken into account and let k be a fixed integer between 0 and p. Writing System (4.1) in the form $\mathcal{D}_k(y(x)) = D(x)\vartheta_k(y(x)) + N(x)\,y(x) = 0$ as explained in the proof of Lemma 4.2.1, we find that $|\alpha(\mathcal{D}_k)|$ is lower than or equal to $n\,p$. Thus, computing a k-simple form of System (4.1) using Algorithm **SimpleForm** can be done in $O\big(\nu\,p\,n^4\big)$ operations in $\mathbb{K}$, while using the super-reduction algorithm [58], it can be done in $O\big(\nu\,p\,n^4\,\min\{n-1,p\}\big)$ operations in $\mathbb{K}$ (see [25]). Consequently, our complexity analysis shows that our algorithm gains the factor $\min\{n-1,p\}$ with respect to the super-reduction. This can be explained by the fact that the super-reduction produces k-simple forms for all $k = 0, \ldots, p$ while our algorithm produces a k-simple form for a single integer k between 0 and p.

4.5 Preservation of the simplicity

We consider a non-simple matrix differential operator of $\mathbb{K}[[x]][\vartheta_k]^{n\times n}$ of the form

$$\mathcal{D}_k = D(x)\,\vartheta_k + N(x),$$

with $D(x) \in \mathcal{A}_n$ and for $i = 1, \ldots, n$, $\min\{v(D(i,.)), v(N(i,.))\} = 0$. We assume that the operator $\mathcal{D}_k$ *written* respectively with ϑ_{k+j}, for $j = 1, 2, \ldots$, is simple (see below). The purpose of this section is to show that Algorithm **SimpleForm** applied to the operator $\mathcal{D}_k$ returns an equivalent k-simple operator without affecting the simplicity with respect to ϑ_{k+j}, for $j = 1, 2, \ldots$. In other terms, the output operator is also simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$.

Let us clarify what we mean by the expressions: *the operator $\mathcal{D}_k$ written with ϑ_{k+j} is simple* or *the operator $\mathcal{D}_k$ is simple with respect to ϑ_{k+j}*. First, we mean by *the operator $\mathcal{D}_k$ written with ϑ_{k+j}* the operator $\mathcal{D}_{k+j}$ derived from $\mathcal{D}_k$ as follows. For $i = 0, \ldots, j$, define the integer n_i by the number of rows of $D(x)$ of valuation i and let $\gamma_i = \sum_{s=0}^{i} n_s$. In particular, n_0 denotes the rank of $D(0)$ unlike the previous sections where it was denoted by r. The operator $\mathcal{D}_{k+j}$ is defined as the operator obtained by multiplying $\mathcal{D}_k$ on the left by the matrix

$$U_j(x) = \operatorname{diag}\big(x^j I_{n_0}, x^{j-1} I_{n_1}, x^{j-2} I_{n_2}, \ldots, x\, I_{n_{j-1}}, I_{n-\gamma_{j-1}}\big)\,. \tag{4.24}$$

We have then

$$\mathcal{D}_{k+j} = U_j(x)\,\mathcal{D}_k = U_j(x)\,D(x)\,\vartheta_k + U_j(x)\,N(x) = D_j(x)\,\vartheta_{k+j} + N_j(x), \tag{4.25}$$

where $N_j(x) = U_j(x)\,N(x)$ and $D_j(x) = x^{-j}U_j(x)\,D(x) \in \mathcal{A}_n$. Thus, we will say that *the operator $\mathcal{D}_k$ written with ϑ_{k+j} is simple* or *the operator $\mathcal{D}_k$ is simple with respect to ϑ_{k+j}* if the operator $\mathcal{D}_{k+j}$ defined in (4.25) is simple with respect to ϑ_{k+j}, that is, if the matrix pencil $D_j(0)\,\lambda + N_j(0)$ is regular. In the sequel, we denote by $L_k(\lambda)$, respectively $L_{k+j}(\lambda)$, the matrix pencil associated with $\mathcal{D}_k$, respectively with $\mathcal{D}_{k+j}$ defined in (4.25). The matrix pencil $L_{k+j}(\lambda)$ is equal to

$$L_{k+j}(\lambda) = D_j(0)\,\lambda + N_j(0) = \operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big)\,\lambda + \operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\,N(0). \tag{4.26}$$

Example 4.5.1. *Consider the non-simple matrix differential operator*

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} x - 2\,x^2 & 1 & 1 & 0 \\ 0 & -x^3 & 0 & -2 \\ 3\,x & 0 & -x-1 & 1 \\ 0 & 0 & 1 & 1-x^3 \end{pmatrix}. \tag{4.27}$$

Here, the integers n_i defined as above are equal to $n_0 = 1$, $n_1 = 2$, $n_2 = 1$ and $n_i = 0$ for $i \geq 3$. Thus, the operator $\mathcal{D}$ written with ϑ_1 is equal to

$$\mathcal{D}_1 = \operatorname{diag}(x,1,1,1)\,\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \vartheta_1 + \begin{pmatrix} x^2 - 2\,x^3 & x & x & 0 \\ 0 & -x^3 & 0 & -2 \\ 3\,x & 0 & -x-1 & 1 \\ 0 & 0 & 1 & 1-x^3 \end{pmatrix}$$

whose associated matrix pencil

$$L_1(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & -2 \\ 0 & 0 & \lambda - 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{4.28}$$

is regular. Hence, the operator $\mathcal{D}$ defined in (4.27) is simple with respect to ϑ_1. Similarly, the operator $\mathcal{D}$ written with ϑ_2 is equal to

$$\mathcal{D}_2 = \operatorname{diag}(x^2,x,x,1)\,\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \vartheta_2 + \begin{pmatrix} x^3 - 2\,x^4 & x^2 & x^2 & 0 \\ 0 & -x^4 & 0 & -2\,x \\ 3\,x^2 & 0 & -x^2 - x & x \\ 0 & 0 & 1 & 1-x^3 \end{pmatrix}.$$

The matrix pencil associated with $\mathcal{D}_2$

$$L_2(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 1 & \lambda + 1 \end{pmatrix} \tag{4.29}$$

is regular and hence the operator $\mathcal{D}$ is also simple with respect to ϑ_2. For $j \geq 3$, Operator $\mathcal{D}$ is always simple with respect to ϑ_j since $L_j(\lambda) = I_4\,\lambda$. ■

Remark 4.5.1. A matrix differential operator $\mathcal{D}_k = D(x)\,\vartheta_k + N(x)$ with $D(x) \in \mathcal{A}_n$ is not necessarily simple with respect to ϑ_{k+j} for every integer $j \geq 1$. For example, the matrix differential operator defined by

$$\mathcal{D}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x^2 & 0 \\ 0 & 0 & x^2 \end{pmatrix} \vartheta_1 + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

is not simple with respect to ϑ_2.

In what follows, we will show that by applying respectively Propositions 4.3.1, 4.3.2 and 4.3.3 to the operator $\mathcal{D}_k$, we do not lose the simplicity with respect to ϑ_{k+j} for $j = 1, 2, \ldots$.

First, we remark that

Lemma 4.5.1. *Given a non-simple matrix differential operator $\mathcal{D}_k = D(x)\,\vartheta_k + N(x)$ with $D(x) \in \mathcal{A}_n$ and assuming that $\mathcal{D}_k$ is simple with respect to ϑ_{k+1}, the integer n_1 defined by the number of rows of $D(x)$ of valuation 1 is necessarily nonzero.*

Proof. We will use the notation above. If $n_1 = 0$, then $n_0 = \gamma_0 = \gamma_1$ and the matrix pencils $L_k(\lambda)$ and $L_{k+1}(\lambda)$ are respectively given by

$$L_k(\lambda) = \operatorname{diag}(I_{n_0}, 0_{n-n_0})\,\lambda + N(0) \quad \text{and} \quad L_{k+1}(\lambda) = \operatorname{diag}(I_{n_0}, 0_{n-n_0})\,\lambda + \operatorname{diag}(0_{n_0}, I_{n-n_0})\,N(0).$$

Since $\mathcal{D}_{k+1}$ is simple ($\mathcal{D}_k$ is simple with respect to ϑ_{k+1}), the regularity of the matrix pencil $L_{k+1}(\lambda)$ implies that the submatrix of $N(0)$ composed of rows and columns of index $n_0 + 1$ to n is invertible. This implies that the matrix pencil $L_k(\lambda)$ is necessarily regular which is impossible. □

We assume first that the constant rows of the matrix pencil $L_k(\lambda)$, *i.e.*, the rows of the matrix $N(0) = N_0$ of index $n_0 + 1$ to n, are linearly dependent. Partition matrix N_0 into blocks N_0^{ij} with $i, j = 0, 1, 2$, where N_0^{00}, N_0^{11} and N_0^{22} are square matrices respectively of size n_0, n_1 and $n - \gamma_1$. Hence, $L_k(\lambda)$ has the block partition

$$L_k(\lambda) = \begin{pmatrix} I_{n_0}\,\lambda + N_0^{00} & N_0^{01} & N_0^{02} \\ N_0^{10} & N_0^{11} & N_0^{12} \\ N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix}$$

with

$$\operatorname{rank} \begin{pmatrix} N_0^{10} & N_0^{11} & N_0^{12} \\ N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix} < n - n_0.$$

Consequently, the matrix pencil $L_{k+1}(\lambda)$ associated with $\mathcal{D}_{k+1}$ has the block partition

$$\begin{aligned} L_{k+1}(\lambda) &= \operatorname{diag}(I_{\gamma_1}, 0_{n-\gamma_1})\,\lambda + \operatorname{diag}(0_{\gamma_0}, I_{n-\gamma_0})\,N(0) \\ &= \begin{pmatrix} I_{n_0}\,\lambda & 0 & 0 \\ N_0^{10} & I_{n_1}\,\lambda + N_0^{11} & N_0^{12} \\ N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix}. \end{aligned}$$

Since $L_{k+1}(\lambda)$ is regular ($\mathcal{D}_k$ is simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$), the submatrix $\begin{pmatrix} N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix}$ is then of full row-rank. Hence, swapping the rows and columns of $\mathcal{D}_k$ of index $n_0 + 1$ to $\gamma_1 = n_0 + n_1$ (this operation does not affect the simplicity with respect to ϑ_{k+j} for $j = 1, 2, \ldots$), we can assume that the first row of

$$\begin{pmatrix} N_0^{10} & N_0^{11} & N_0^{12} \\ N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix}$$

can be written as a linear combination of the other rows.

Proposition 4.5.1. *Let $\mathcal{D}_k$ be a non-simple matrix differential operator having a leading coefficient matrix in $\mathcal{A}_n$. Assume that $\mathcal{D}_k$ is simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$. Moreover, assume that the row of $L_k(\lambda)$ of index $n_0 + 1$ can be written as a linear combination of the rows of index $n_0 + 2$ to n. Let $\widetilde{\mathcal{D}}_k = \widetilde{D}(x)\,\vartheta_k + \widetilde{N}(x)$ denote the operator obtained after applying the process described in the proof of Proposition 4.3.1 to $\mathcal{D}_k$. Then, $\widetilde{\mathcal{D}}_k$ is also simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$.*

Proof. The two operators $\widetilde{\mathcal{D}}_k$ and $\mathcal{D}_k$ are related by $\widetilde{\mathcal{D}}_k = S(x)\,\mathcal{D}_k\,T$, where $S(x)$ is of the form

$$S(x) = \operatorname{diag}\big(I_{n_0}, x^{-1}, I_{n_1-1}, I_{n-\gamma_1}\big)\,S_1$$

with

$$S_1 = \left(\begin{array}{c|cc|c} I_{n_0} & 0 & 0 & 0 \\ \hline 0 & 1 & u & v \\ 0 & 0 & I_{n_1-1} & 0 \\ \hline 0 & 0 & 0 & I_{n-\gamma_1} \end{array}\right), \tag{4.30}$$

where $u \in \mathbb{K}^{1\times(n_1-1)}$, $v \in \mathbb{K}^{1\times(n-\gamma_1)}$ and the row vector $\begin{pmatrix} 1 & u & v \end{pmatrix}$ belongs to the left nullspace of matrix

$$\begin{pmatrix} N_0^{10} & N_0^{11} & N_0^{12} \\ N_0^{20} & N_0^{21} & N_0^{22} \end{pmatrix},$$

and T is given by

$$T = \left(\begin{array}{c|cc|c} I_{n_0} & 0 & 0 & 0 \\ \hline 0 & 1 & -u & 0 \\ 0 & 0 & I_{n_1-1} & 0 \\ \hline 0 & 0 & 0 & I_{n-\gamma_1} \end{array}\right). \tag{4.31}$$

Let $\widetilde{\mathcal{D}}_{k+j}$ denote the operator $\widetilde{\mathcal{D}}_k$ written with ϑ_{k+j}. We have $\widetilde{\mathcal{D}}_{k+j} = \widetilde{U}_j(x)\,\widetilde{\mathcal{D}}_k$ with

$$\widetilde{U}_j(x) = \operatorname{diag}\big(x^j I_{\widetilde{n}_0}, x^{j-1} I_{\widetilde{n}_1}, x^{j-2} I_{n_2}, \ldots, x\, I_{n_{j-1}}, I_{n-\gamma_{j-1}}\big),$$

where the integers $\widetilde{n}_i$ denote the number of rows of $\widetilde{D}(x)$ of valuation i. Notice that $\widetilde{n}_0 = n_0+1$, $\widetilde{n}_1 = n_1 - 1$ and $\widetilde{n}_i = n_i$ for $i \geq 2$. Write

$$\widetilde{\mathcal{D}}_{k+j} = \widetilde{U}_j(x)\,\widetilde{\mathcal{D}}_k = \widetilde{U}_j(x)\,\widetilde{D}(x)\,\vartheta_k + \widetilde{U}_j(x)\,\widetilde{N}(x) = \widetilde{D}_j(x)\,\vartheta_{k+j} + \widetilde{N}_j(x),$$

where $\widetilde{D}_j(x) = x^{-j}\widetilde{U}_j(x)\,\widetilde{D}(x) \in \mathcal{A}_n$ and

$$\begin{aligned} \widetilde{N}_j(x) = \widetilde{U}_j(x)\,\widetilde{N}(x) &= \widetilde{U}_j(x)\,S(x)\,N(x)\,T \\ &= \underbrace{\widetilde{U}_j(x)\operatorname{diag}\big(I_{n_0}, x^{-1}, I_{n_1-1}, I_{n-\gamma_1}\big)}_{U_j(x)} S_1\,N(x)\,T \\ &= U_j(x)\,S_1\,N(x)\,T, \end{aligned}$$

where $U_j(x)$, S_1 and T are the matrices respectively given in (4.24), (4.30) and (4.31). Let $\widetilde{L}_{k+j}(\lambda)$ denote the matrix pencil associated with $\widetilde{\mathcal{D}}_{k+j}$. We will show that $\widetilde{L}_{k+j}(\lambda)$ is regular for $j \geq 1$. For this, we will distinguish two cases: $j = 1$ and $j \geq 2$. For $j = 1$, we have

$$\widetilde{L}_{k+1}(\lambda) = \widetilde{D}_1(0)\,\lambda + \widetilde{N}_1(0) = \operatorname{diag}(I_{\gamma_1}, 0_{n-\gamma_1})\,\lambda + \operatorname{diag}(0_{n_0}, I_{n-n_0})\,S_1\,N(0)\,T.$$

Notice that

$$\operatorname{diag}(0_{n_0}, I_{n-n_0})\,S_1 = S_1\operatorname{diag}(0_{n_0}, I_{n-n_0}) \quad \text{and} \quad \operatorname{diag}(I_{\gamma_1}, 0_{n-\gamma_1}) = S_1\operatorname{diag}(I_{\gamma_1}, 0_{n-\gamma_1})\,T.$$

Thus, $\widetilde{L}_{k+1}(\lambda)$ can be written as

$$\widetilde{L}_{k+1}(\lambda) = S_1\operatorname{diag}(I_{\gamma_1}, 0_{n-\gamma_1})\,T\,\lambda + S_1\operatorname{diag}(0_{n_0}, I_{n-n_0})\,N(0)\,T = S_1\,L_{k+1}(\lambda)\,T,$$

where $L_{k+1}(\lambda)$ is the matrix pencil given in (4.26) for $j = 1$. Since $L_{k+1}(\lambda)$ is regular ($\mathcal{D}_k$ is simple with respect to ϑ_{k+1}), it follows that $\widetilde{L}_{k+1}(\lambda)$ is regular as well and the operator $\widetilde{\mathcal{D}}_k$ is simple with respect to ϑ_{k+1}. For $j \geq 2$, we have

$$\widetilde{L}_{k+j}(\lambda) = \widetilde{D}_j(0)\,\lambda + \widetilde{N}_j(0) = \operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big)\,\lambda + \operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\,S_1\,N(0)\,T$$

with $\gamma_j \geq \gamma_{j-1} \geq \gamma_1 = n_0 + n_1 = \widetilde{n}_0 + \widetilde{n}_1 \geq n_0 + 1$. Thus, we have

$$\begin{aligned} &\operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\, S_1 = \operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big) = T^{-1}\operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big) \\ \text{and} \qquad &\operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big) = T^{-1}\operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big)\, T. \end{aligned}$$

So, we have

$$\widetilde{D}_j(0) = \operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big) = T^{-1}\operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big)\, T$$

and

$$\begin{aligned} \widetilde{N}_j(0) = \operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\, S_1\, N(0)\, T &= \operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\, N(0)\, T \\ &= T^{-1}\operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\, N(0)\, T. \end{aligned}$$

Consequently, $\widetilde{L}_{k+j}(\lambda)$ and $L_{k+j}(\lambda)$ are related by

$$\widetilde{L}_{k+j}(\lambda) = T^{-1}\operatorname{diag}\big(I_{\gamma_j}, 0_{n-\gamma_j}\big)\, T\,\lambda + T^{-1}\operatorname{diag}\big(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}\big)\, N(0)\, T = T^{-1} L_{k+j}(\lambda)\, T.$$

Thus $\widetilde{L}_{k+j}(\lambda)$ is regular since L_{k+j} is so, and the operator $\widetilde{\mathcal{D}}_k$ is simple with respect to ϑ_{k+j}. □

Example 4.5.2. *Consider the operator $\mathcal{D}$ given in (4.27). Its associated matrix pencil is given by*

$$L(\lambda) = \begin{pmatrix} \lambda & 1 & 1 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Remark that we have $L(2,.) = -L(3,.) - L(4,.)$. Hence, following the process described in the proof of Proposition 4.3.1, we construct two matrices $S(x) = \operatorname{diag}\big(1, x^{-1}, 1, 1\big)\, S_1$ with

$$S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \textit{and} \quad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

such that $\widetilde{\mathcal{D}} = S(x)\,\mathcal{D}\,T$ given by

$$\widetilde{\mathcal{D}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & x \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} x - 2\,x^2 & 1 & 0 & 0 \\ 3 & -x^2 & x^2 - 1 & -x^2 \\ 3\,x & 0 & -x - 1 & 1 \\ 0 & 0 & 1 & 1 - x^3 \end{pmatrix}$$

satisfies $\Big|\alpha(\widetilde{\mathcal{D}})\Big| = |\alpha(\mathcal{D})| - 1$. Operator $\widetilde{\mathcal{D}}$ written with ϑ_1 in the form

$$\widetilde{\mathcal{D}}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \vartheta_1 + \begin{pmatrix} x^2 - 2\,x^3 & x & 0 & 0 \\ 3\,x & -x^3 & x^3 - x & -x^3 \\ 3\,x & 0 & -x - 1 & 1 \\ 0 & 0 & 1 & 1 - x^3 \end{pmatrix}$$

has an associated matrix pencil equal to

$$\widetilde{L}_1(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda - 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

One can easily check that $\widetilde{L}_1(\lambda) = S_1\, L_1(\lambda)\, T$, *where* $L_1(\lambda)$ *is given by (4.28). Similarly, operator* $\widetilde{\mathcal{D}}$ *written with* ϑ_2 *has an associated matrix pencil*

$$\widetilde{L}_2(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 1 & \lambda + 1 \end{pmatrix}$$

with $\widetilde{L}_2(\lambda) = T^{-1}\, L_2(\lambda)\, T$, *where* $L_2(\lambda)$ *is given by (4.29). For* $j \geq 3$, *we have* $\widetilde{L}_j(\lambda) = I_4\, \lambda = T^{-1}\, L_j(\lambda)\, T$ *since* $L_j(\lambda) = I_4\, \lambda$. ■

Proposition 4.5.2. *Let* $\mathcal{D}_k$ *be a non-simple matrix differential operator having a leading coefficient matrix in* $\mathcal{A}_n$. *Assume that* $\mathcal{D}_k$ *is simple with respect to* ϑ_{k+j} *for* $j = 1, 2, \ldots$. *Furthermore, assume that the matrix pencil* $L_k(\lambda)$ *associated with* $\mathcal{D}_k$ *is in the form (4.15) with condition (4.16) satisfied. Let* $\widetilde{\mathcal{D}}_k = S(x)\, \mathcal{D}_k\, T(x)$, *where* $S(x)$ *and* $T(x)$ *are respectively given by (4.17) and (4.18) in Proposition 4.3.2. Then,* $\widetilde{\mathcal{D}}_k$ *is simple with respect to* ϑ_{k+j} *for* $j = 1, 2, \ldots$.

Proof. We will show that $\widetilde{\mathcal{D}}_k$ is simple with respect to ϑ_{k+1}. The simplicity of $\widetilde{\mathcal{D}}_k$ with respect to ϑ_{k+j} for $j \geq 2$ can be proved in a similar way. For the sake of clarity, we will partition $L_k(\lambda)$ into blocks as follows

$$L_k(\lambda) = \begin{pmatrix} I_q\lambda + N_0^{00} & 0 & 0 & 0 \\ N_0^{10} & I_{n_0-q}\lambda + N_0^{11} & N_0^{12} & N_0^{13} \\ N_0^{20} & N_0^{21} & N_0^{22} & N_0^{23} \\ N_0^{30} & N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix},$$

where $0 \leq q \leq n_0$, N_0^{22} and N_0^{33} are square matrices respectively of size n_1 and $n - \gamma_1$ and

$$\operatorname{rank} \begin{pmatrix} N_0^{21} & N_0^{22} & N_0^{23} \\ N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix} < n - n_0.$$

In this case, the matrix pencil $L_{k+1}(\lambda)$ given by (4.26) for $j = 1$ (associated with the operator $\mathcal{D}_{k+1}$ given by (4.25)) is of the form

$$L_{k+1}(\lambda) = \begin{pmatrix} I_q\lambda & 0 & 0 & 0 \\ 0 & I_{n_0-q}\lambda & 0 & 0 \\ N_0^{20} & N_0^{21} & I_{n_1}\lambda + N_0^{22} & N_0^{23} \\ N_0^{30} & N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix}.$$

Due to its particular block structure, the regularity of $L_{k+1}(\lambda)$ implies that the submatrix

$$\begin{pmatrix} I_{n_1}\lambda + N_0^{22} & N_0^{23} \\ N_0^{32} & N_0^{33} \end{pmatrix}$$

is regular as well. Now, the matrix pencil associated with the operator $\widetilde{\mathcal{D}}_k = S(x)\, \mathcal{D}_k\, T(x)$ is of the form

$$\widetilde{L}_k(\lambda) = \begin{pmatrix} I_q\lambda + * & * & * & * \\ 0 & I_{n_0-q}\lambda + N_0^{11} & N_0^{12} & N_0^{13} \\ 0 & N_0^{21} & N_0^{22} & N_0^{23} \\ 0 & N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix},$$

where the $*$ denotes the remaining blocks of the matrix. Since $\alpha(\mathcal{D}_k) = \alpha(\widetilde{\mathcal{D}}_k)$, the matrix pencil associated with the operator $\widetilde{\mathcal{D}}_k$ written with ϑ_{k+1} is given by

$$\widetilde{L}_{k+1}(\lambda) = \begin{pmatrix} I_q\lambda & 0 & 0 & 0 \\ 0 & I_{n_0-q}\lambda & 0 & 0 \\ 0 & N_0^{21} & I_{n_1}\lambda + N_0^{22} & N_0^{23} \\ 0 & N_0^{31} & N_0^{32} & N_0^{33} \end{pmatrix}$$

and hence it is regular since the submatrix

$$\begin{pmatrix} I_{n_1}\lambda + N_0^{22} & N_0^{23} \\ N_0^{32} & N_0^{33} \end{pmatrix}$$

remained intact. □

Proposition 4.5.3. *Consider a non-simple matrix differential operator $\mathcal{D}_k$ having a leading coefficient matrix in $\mathcal{A}_n$. Assume that $\mathcal{D}_k$ is simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$. Let S and T be the two invertible constant matrices computed as in the proof of Proposition 4.3.3 such that the matrix pencil associated with $\widetilde{\mathcal{D}}_k = S\,\mathcal{D}_k\,T$ is of the form (4.15) with condition (4.16) satisfied. Then, $\widetilde{\mathcal{D}}_k$ is simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$.*

Proof. From the proof of Proposition 4.3.3, we deduce that the matrices S and T are of the form

$$S = \begin{pmatrix} S_1 & S_2 \\ 0 & I_{n-n_0} \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} S_1^{-1} & 0 \\ 0 & I_{n-n_0} \end{pmatrix},$$

where $S_1 \in \mathrm{GL}_{n_0}(\mathbb{K})$ and $S_2 \in \mathbb{K}^{n_0\times(n-n_0)}$. Put $\widetilde{\mathcal{D}}_k = S\,\mathcal{D}_k\,T = \widetilde{D}(x)\,\vartheta_k + \widetilde{N}(x)$, where $\widetilde{D}(x) = S\,D(x)\,T$ and $\widetilde{N}(x) = S\,N(x)\,T$. Let $\widetilde{\mathcal{D}}_{k+j}$ denote the operator $\widetilde{\mathcal{D}}_k$ written with ϑ_{k+j}. Since $\alpha(\widetilde{\mathcal{D}}_k) = \alpha(\mathcal{D}_k)$, then we have $\widetilde{\mathcal{D}}_{k+j} = U_j(x)\,\widetilde{\mathcal{D}}_k$ with $U_j(x)$ defined in (4.24). Write $\widetilde{\mathcal{D}}_{k+j} = \widetilde{D}_j(x)\,\vartheta_{k+j} + \widetilde{N}_j(x)$ with $\widetilde{D}_j(x) = x^{-j}U_j(x)\,\widetilde{D}(x) \in \mathcal{A}_n$ and $\widetilde{N}_j(x) = U_j(x)\,\widetilde{N}(x)$. The matrix pencil associated with $\widetilde{\mathcal{D}}_{k+j}$ is equal to

$$\widetilde{L}_{k+j}(\lambda) = \widetilde{D}_j(0)\,\lambda + \widetilde{N}_j(0) = \mathrm{diag}(I_{\gamma_j}, 0_{n-\gamma_j})\,\lambda + \mathrm{diag}(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}})\,S\,N(0)\,T.$$

As $\gamma_j \geq \gamma_{j-1} \geq \gamma_0 = n_0$, it follows that

$$\mathrm{diag}(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}})\,S = \mathrm{diag}(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}}) = T^{-1}\mathrm{diag}(0_{\gamma_{j-1}}, I_{n-\gamma_{j-1}})$$

and

$$\mathrm{diag}(I_{\gamma_j}, 0_{n-\gamma_j}) = T^{-1}\,\mathrm{diag}(I_{\gamma_j}, 0_{n-\gamma_j})\,T.$$

Thus, we have $\widetilde{L}_{k+j}(\lambda) = T^{-1}\,L_{k+j}(\lambda)\,T$, where $L_{k+j}(\lambda)$ is the matrix pencil given by (4.26) associated with $\mathcal{D}_{k+j}$. Hence, the operator remains simple with respect to ϑ_{k+j} for $j \geq 1$. □

Example 4.5.3. *Consider the non-simple matrix differential operator given by*

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{pmatrix}\vartheta + \begin{pmatrix} 1+x & 0 & 2\,x & 0 \\ 1 & 1 & x^2+x^3 & 0 \\ 0 & 5\,x & 1 & 1 \\ 0 & 1 & 0 & x^3+2\,x \end{pmatrix}.$$

The matrix pencil associated with the operator $\mathcal{D}$ written with ϑ_j (for $j \geq 1$) is equal to

$$L_j(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda+1 & 1 \\ 0 & 1 & 0 & \lambda \end{pmatrix} \quad \text{if } j = 1,$$

and $L_j(\lambda) = I_4\,\lambda$ if $j \geq 2$. Hence, $\mathcal{D}$ is simple with respect to ϑ_j for $j \geq 1$. Following the construction in the proof of Proposition 4.3.3, we get two matrices

$$S = \left(\begin{array}{cc|cc} 1 & -1 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \quad \text{and} \quad T = \left(\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

such that the operator

$$\widetilde{\mathcal{D}} = S\,\mathcal{D}\,T = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & -2\,x \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{pmatrix} \vartheta + \begin{pmatrix} x & x & 2\,x - x^2 - x^3 & x^3 + 2\,x \\ 1 & 0 & x^2 + x^3 & -2\,x^3 - 4\,x \\ 0 & 5\,x & 1 & 1 \\ 0 & 1 & 0 & x^3 + 2\,x \end{pmatrix}$$

has an associated matrix pencil of the form (4.15) with $q = 2$ and condition (4.16) satisfied. The matrix pencil associated with the operator $\widetilde{\mathcal{D}}$ written with ϑ_j is equal to

$$\widetilde{L}_j(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda + 1 & 1 \\ 0 & 1 & 0 & \lambda \end{pmatrix} \quad \text{if } j = 1,$$

and $\widetilde{L}_j(\lambda) = I_4\,\lambda$ if $j \geq 2$. One can easily check that the relation $\widetilde{L}_j(\lambda) = T^{-1}\,L_j(\lambda)\,T$ for $j \geq 1$ is well satisfied. ■

By combining the results of Propositions 4.5.1, 4.5.3 and 4.5.2, we obtain the following theorem.

Theorem 4.5.1. *Let $\mathcal{D}_k = D(x)\,\vartheta_k + N(x) \in \mathbb{K}[[x]][\vartheta_k]^{n\times n}$ be a matrix differential operator having a leading coefficient matrix $D(x)$ in $\mathcal{A}_n$. If $\mathcal{D}_k$ is simple with respect to ϑ_{k+j} for $j = 1, 2, \ldots$, then Algorithm* **SimpleForm** *applied to $\mathcal{D}_k$ returns an operator which is simple with respect to ϑ_k and to ϑ_{k+j} for $j = 1, 2, \ldots$.*

4.6 An example

We end with an example recapitulating all the notions viewed in this chapter.

Consider the system of first-order linear differential equations given by

$$\vartheta(y(x)) + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 - x^2 & 0 & 0 & 0 & -x \\ -\frac{1}{x^2} + 1 & 0 & 0 & \frac{1}{x^3} - \frac{1}{x} & \frac{2}{x} \end{pmatrix} y(x) = 0 \tag{4.32}$$

having a Poincaré-rank equal to 3. The slopes of the Newton polygon of System (4.32) are then rational numbers between 0 and 3. Write System (4.32) in the form

$$\mathcal{D}(y(x)) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x^3 \end{pmatrix} \vartheta(y(x)) + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 - x^2 & 0 & 0 & 0 & -x \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x^2 \end{pmatrix} y(x) = 0. \tag{4.33}$$

In the sequel, we would like to compute an operator equivalent to $\mathcal{D}$ which is simple with respect to ϑ_i for $i = 0, 1, 2, 3$ and hence determine all integer slopes of the Newton polygon of (4.32). Remark first that the operator $\mathcal{D}$ written respectively with ϑ_1 and ϑ_2 is not simple, but written

with ϑ_3, it is so. Secondly, Algorithm **SimpleForm** applied to $\mathcal{D}$ given in (4.33) returns the operator

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -x^2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x^2 \end{pmatrix} \vartheta + \begin{pmatrix} x - x^3 + 1 & x^2 & 0 & 0 & -2\,x \\ 1 - x^2 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ x^3 - x & 1 - x^2 & 0 & 0 & 2\,x \end{pmatrix}.$$

which is simple with respect to ϑ but neither simple with respect to ϑ_1 nor with respect to ϑ_2. To achieve our purpose, we will proceed as follows. Operator $\mathcal{D}$ written with ϑ_3 is equal to

$$\mathcal{D}_3 = D_3(x)\,\vartheta_3 + N_3(x) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \vartheta_3 + \begin{pmatrix} 0 & 0 & 0 & x^3 & 0 \\ 0 & 0 & 0 & 0 & x^3 \\ 0 & x^3 & 0 & 0 & 0 \\ x^3 - x^5 & 0 & 0 & 0 & -x^4 \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x^2 \end{pmatrix}$$

and it is simple with respect to ϑ_3 since the determinant of its associated matrix pencil is equal to λ^5 (this implies that 3 is not a slope of the Newton polygon of (4.32) since the determinant has only 0 as root). Write $\mathcal{D}_3$ with ϑ_2 ($\vartheta_3 = x\,\vartheta_2$) as follows $\mathcal{D}_3 = x\,D_3(x)\,\vartheta_2 + N_3(x)$, then simplify by x the rows of index 1 to 4 of the operator so that all rows of the operator can be of valuation 0. We find

$$\mathcal{D}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x \end{pmatrix} \vartheta_2 + \begin{pmatrix} 0 & 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & 0 & x^2 \\ 0 & x^2 & 0 & 0 & 0 \\ x^2 - x^4 & 0 & 0 & 0 & -x^3 \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x^2 \end{pmatrix}$$

which is not simple with respect to ϑ_2 but simple with respect to ϑ_3 ($\mathcal{D}_2$ written with ϑ_3 is the operator $\mathcal{D}_3$ which is simple with respect to ϑ_3). Applying Algorithm **SimpleForm** to $\mathcal{D}_2$, we find the equivalent simple operator

$$\widetilde{\mathcal{D}}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \vartheta_2 + \begin{pmatrix} x^2 & 0 & 0 & x^2 & 0 \\ 0 & x^2 & 0 & 0 & x \\ 0 & x^2 & x^2 & 0 & 0 \\ x^2 - x^4 & 0 & 0 & x^2 & -x^2 \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x \end{pmatrix}$$

which is simple with respect to respectively ϑ_2 and ϑ_3 (see Theorem 4.5.1). The determinant of the matrix pencil associated with $\widetilde{\mathcal{D}}_2$ is equal to λ^5 and hence the integer 2 is not a slope of the Newton polygon of (4.32). Since $\vartheta_2 = x\,\vartheta_1$, the operator $\widetilde{\mathcal{D}}_2$ can be written with ϑ_1 as follows

$$\begin{pmatrix} x & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x \end{pmatrix} \vartheta_1 + \begin{pmatrix} x^2 & 0 & 0 & x^2 & 0 \\ 0 & x^2 & 0 & 0 & x \\ 0 & x^2 & x^2 & 0 & 0 \\ x^2 - x^4 & 0 & 0 & x^2 & -x^2 \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x \end{pmatrix}.$$

Simplifying the rows of index 1 to 4 of the latter operator by x, we find the operator

$$\mathcal{D}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x \end{pmatrix} \vartheta_1 + \begin{pmatrix} x & 0 & 0 & x & 0 \\ 0 & x & 0 & 0 & 1 \\ 0 & x & x & 0 & 0 \\ x - x^3 & 0 & 0 & x & -x \\ x^3 - x & 0 & 0 & 1 - x^2 & 2\,x \end{pmatrix}$$

which is simple with respect to ϑ_2 and ϑ_3 (since $\widetilde{\mathcal{D}}_2$ is so) but not simple with respect to ϑ_1. Again, by applying Algorithm **SimpleForm** to $\mathcal{D}_1$, we obtain the equivalent operator

$$\widetilde{\mathcal{D}}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \vartheta_1 + \begin{pmatrix} 2\,x & 0 & x & 0 & 0 \\ 0 & 2\,x & 0 & 1 & 0 \\ x - x^3 & 0 & 2\,x & 0 & -1 \\ 0 & 0 & 0 & x & 1 \\ x^3 - x & 0 & 1 - x^2 & 0 & 2 \end{pmatrix}$$

which is simple with respect to ϑ_1, ϑ_2 and ϑ_3. The determinant of the matrix pencil associated with $\widetilde{\mathcal{D}}_1$ is equal to $\lambda^3\,(\lambda+1)^2$ which means that 1 is a slope of the Newton Polygon and System (4.32) has two irregular solutions of the form

$$y(x) = \exp\left(\int \frac{-1}{x^2} + \dots\right) z(x),$$

where the dots stand for terms of valuation higher than -2 and $z(x) \in \overline{\mathbb{K}}[[x^{1/r}]]^5[\log(x)]$ with $r \in \mathbb{N}^*$. Now write $\widetilde{\mathcal{D}}_1$ with ϑ. After simplifying the first row of the operator by x, we find

$$\mathcal{D}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & x \end{pmatrix} \vartheta + \begin{pmatrix} 2 & 0 & 1 & 0 & 0 \\ 0 & 2\,x & 0 & 1 & 0 \\ x - x^3 & 0 & 2\,x & 0 & -1 \\ 0 & 0 & 0 & x & 1 \\ x^3 - x & 0 & 1 - x^2 & 0 & 2 \end{pmatrix}.$$

Now, Algorithm **SimpleForm** applied to $\mathcal{D}_0$ gives rise to the operator $\widetilde{\mathcal{D}}_0 = \widetilde{D}(x)\,\vartheta + \widetilde{N}(x)$, where

$$\widetilde{D}(x) = \operatorname{diag}(1, 1, 1, x, x)$$

and

$$\widetilde{N}(x) = \begin{pmatrix} 2\,x^2 + 3\,x^3 - 3\,x + 1 & x\left(x + 6\,x^2 - 6\right) & -x^2 + 12\,x^3 - 12\,x + 2 & 4 + 2\,x^2 - 6\,x^3 + 9\,x & -5 - x^2 + 3\,x^3 - 3\,x \\ 1 - x^2 - 4\,x^3 + 4\,x & 2\,x^2 - 8\,x^3 + 8\,x + 1 & -5 + 8\,x^2 - 16\,x^3 + 16\,x & -4 - 6\,x^2 + 8\,x^3 - 12\,x & 6 + 3\,x^2 - 4\,x^3 + 4\,x \\ x^3 - x & 2\,x^3 - 2\,x + 1 - x^2 & 5 - 4\,x + 4\,x^3 - 3\,x^2 & -x\left(-3 + 2\,x^2 - 2\,x\right) & -1 - x + x^3 - x^2 \\ 0 & 0 & 0 & x & 1 \\ x^2\left(-1 + x^2\right) & 2\,x^4 - 2\,x^2 + x - x^3 & -4\,x^2 + 4\,x^4 + 3\,x - 3\,x^3 & -1 + 3\,x^2 - 2\,x^4 - 2\,x + 2\,x^3 & 2 - x^2 + x^4 + x - x^3 \end{pmatrix}$$

which is simple with respect to ϑ_i for $i = 0, 1, 2, 3$. The determinant of the matrix pencil associated with $\widetilde{\mathcal{D}}_0$ is equal to $(\lambda+2)^2\,(\lambda+3)$. Hence System (4.32) has 5 formal solutions, two of them are irregular solutions and the three others are regular. In this example, the Newton polygon of System (4.32) have only integer slopes.

Conclusion and Perspectives of Part I

In this part of the thesis, we have developed direct approaches for computing regular formal solutions of linear differential systems of arbitrary order in a neighborhood of a singularity. Our algorithms treat simple systems and are in particular valid for systems of the first kind. Thus, the problem of computing regular solutions of higher-order simple linear differential systems has been completely solved.

For non simple linear differential systems, we have proposed two algorithms computing auxiliary simple linear differential systems from which one can recover the regular solutions of the non simple ones. The first algorithm **LinSubs** computes a linear substitution $y(x) = T(x)\, z(x)$ with invertible matrix $T(x)$, when it exists, yielding a simple system. The second algorithm **EG_DV** is a differential variant of the EG'-algorithm [4]. The latter algorithm presents two essential drawbacks: the first is that it only works on systems with polynomial coefficients and the second is that the order of the output system is, in general, higher than that of the input one. It would be interesting in the future to look for algorithms treating the general case, *i.e.*, systems with formal power series coefficients, and computing equivalent simple systems without increasing the order.

Then, we were interested in computing k-simple differential systems of first-order. The k-simple forms are useful in the determination of the integer slopes of the Newton polygon of systems of the form $\vartheta(y(x)) = A(x)\, y(x)$ with $A(x) \in \mathbb{K}((x))^{n\times n}$ and in the computation of the regular solutions and rational solutions [14] as well. The classical approach (before this thesis) for computing k-simple forms is using the super-reduction algorithm [58]. Here, we have proposed a direct method for their computation. While our approach allows to classify the singularity as a regular or an irregular one, some questions concerning it remain however hung: what can we say about the Katz invariant of the system? How, by taking interest of this approach, we can determine the nature (integer or not) of the Katz invariant? What is the analogue, in our approach, of the Moser-reduction [77] regarding the Katz invariant? In other words, how, by using our approach, we can find the smallest integer greater than the Katz invariant? We plan to address these questions in our future researches.

Note that the algorithms presented in this part have been all implemented in Maple[5] and a study of their arithmetic complexities have been provided.

In the future, we intend to investigate a possibility of developing direct algorithms handling higher-order linear differential systems and computing the other type of formal solutions at a singularity, the irregular solutions. A first step toward this would be the generalization of the concept of k-simple forms to higher-order differential systems and the development of efficient algorithms for their computations. We are interested as well in the global analysis of linear differential systems of higher-orders, such as proposing direct methods for computing rational and polynomial solutions. We have already considered this problem and we have noted that the method in [14] can be generalized to higher-order differential systems. But, a prerequisite condition for this method is to have simple forms, so we have to solve the problem of simplicity first. Another direction could be to focus on the problems of divisibility and factorization of higher-order matrix differential operators.

[5]The programs are available at `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`.

Part II

Reduction Algorithms for Linear Differential-Algebraic Equations

Chapter 5

Reduction Algorithms for Linear Differential-Algebraic Equations of First-Order

Contents

5.1 Introduction

In this chapter, we will focus on the study of linear differential-algebraic equations (DAEs) of first-order of the form

$$L(y(x)) = A(x)\,y'(x) + B(x)\,y(x) = f(x), \tag{5.1}$$

where x is a complex variable, $A(x)$ and $B(x)$ are $m \times n$ matrices of analytic functions, $y(x)$ is an unknown n-dimensional vector and the right-hand side $f(x)$ is a m-dimensional vector of analytic functions. We are interested in the local analysis of such systems at the point $x = 0$, and therefore we can suppose, without loss of generality, that the entries of $A(x)$, $B(x)$ and $f(x)$ are formal power series at $x = 0$.

The aim is to develop an algorithm which computes a system equivalent to (5.1) to which the classical theory of ordinary differential equations (ODEs) is applicable.

Linear DAEs of the form (5.1) have been intensively studied from a numerical point of view. Traditionally, they have been tackled using the notion of *differential index* [46] (a range of alternative index definitions exist as well, see for example [67, 89] and references therein). Generally speaking, most authors try to extract the *underlying ODE* which is a system of ordinary differential equations of first-order expressing y' in terms of y and x, computed by differentiating (5.1) successively and then using only algebraic manipulations. The number of differentiations of the initial DAE required to generate the underlying ODE is called the *differential index*.

In [67], the authors considered DAEs of the form (5.1) with continuous coefficient matrices on a real closed interval. They proposed a set of local characterizing quantities and developed an algorithm reducing (5.1) to an equivalent[1] one, with a very special structure, allowing to answer the questions on the existence and the uniqueness of solutions of the initial DAE. Another algorithm, improving the one of [67] and treating the DAEs on all their definition interval, has been presented in [87, 88]. However, the size of the system during execution of the latter algorithm might increase.

In [52], the authors considered DAEs of the form (5.1) with coefficient matrices holomorphic at $x = 0$ and were interested in the existence of solutions of such systems and in the number of solutions which are holomorphic at $x = 0$ or have at most a pole at $x = 0$. They developed an algebraic algorithm reducing the DAE (5.1) to a sequence of first-order systems of ODEs and algebraic systems of lower sizes with some necessary conditions on the right-hand side. We will review this algorithm in more details in Section 5.2.

In this chapter, motivated by the work of Harris et al. [52], we use a different strategy: using the terminology of matrix differential operators, we compute a sequence of left- and right-transformations in order to obtain a new operator having decoupled differential and algebraic parts. Our contribution is then reflected in the development and the implementation in Maple of a new reduction algorithm which, for a given operator $L = A(x)\,\partial + B(x) \in \mathbb{K}[[x]][\partial]^{m\times n}$ with $\partial = \frac{d}{dx}$, returns an operator $\widetilde{L} = S\,L\,T$ of the form

$$\widetilde{L} = \begin{pmatrix} \widetilde{A}_{11}(x)\,\partial + \widetilde{B}_{11}(x) & 0 & 0 \\ 0 & \widetilde{B}_{22}(x) & 0 \\ 0 & 0 & 0 \end{pmatrix} \in \mathbb{K}[[x]][\partial]^{m\times n}, \tag{5.2}$$

where S and T are two unimodular matrix differential operators, and $\widetilde{A}_{11}$ and $\widetilde{B}_{22}$ are invertible matrices over $\mathbb{K}((x))$. Hence, the DAE (5.1) is reduced to $\widetilde{L}(z(x)) = \widetilde{f}(x)$, where $y(x) = T(z(x))$ and $\widetilde{f}(x) = S(f(x))$. Decomposing $z(x)$ and $\widetilde{f}(x)$ into blocks of the same partition as $\widetilde{L}$ in (5.2), *i.e.*,

$$z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ z_3(x) \end{pmatrix} \quad \text{and} \quad \widetilde{f}(x) = \begin{pmatrix} \widetilde{f}_1(x) \\ \widetilde{f}_2(x) \\ \widetilde{f}_3(x) \end{pmatrix},$$

$\widetilde{L}(z(x)) = \widetilde{f}(x)$ can then be written as two separate problems, possibly of lower size than (5.1):

1. one being purely differential: $\widetilde{A}_{11}(x)\,z_1'(x) + \widetilde{B}_{11}(x)\,z_1(x) = \widetilde{f}_1(x)$,
2. and the other one being purely algebraic: $\widetilde{B}_{22}(x)\,z_2(x) = \widetilde{f}_2(x)$,

together with some necessary conditions on the right-hand side expressed by $\widetilde{f}_3(x) = 0$. We conclude the chapter by exploring the notion of singularities associated with system (5.1) (see also [87]).

The content of this chapter constitutes a part of the published paper [20]. The second part of [20] concerning the second-order case has been omitted since linear DAEs of arbitrary order will be treated in the next chapter.

Notation and Terminology. The notation ∂ stands for the standard derivation $\frac{d}{dx}$ of $\mathbb{K}((x))$ ($\mathbb{K}$ being a subfield of $\mathbb{C}$). For $M \in \mathbb{K}((x))^{m\times n}$, we will use sometimes the notation M' for $\frac{dM}{dx}$. In this chapter and the next one, we are interested in applying to an arbitrary matrix differential

[1] Equivalence means that there is one-to-one correspondence of their solutions.

operator L elementary row operations and elementary column operations of the second type (see Subsection 3.6.1 of Chapter 3). From now on, we omit the expression "of the second type" for sake of brevity.

We recall that a square matrix differential operator S of $\mathbb{K}((x))[\partial]^{m\times m}$ is unimodular (*i.e.*, has a two sided-inverse in $\mathbb{K}((x))[\partial]^{m\times m}$) if it can be expressed as a product of elementary operations. Finally, two matrix differential operators L and $\widetilde{L}$ of $\mathbb{K}((x))[\partial]^{m\times n}$ are said to be *equivalent* if there exist two unimodular matrix differential operators $S \in \mathbb{K}((x))[\partial]^{m\times m}$ and $T \in \mathbb{K}((x))[\partial]^{n\times n}$such that $\widetilde{L} = S\,L\,T$.

5.2 Review of Harris et al. algorithm

The algorithm proposed by Harris et al. in [52] reduces a non-homogeneous linear differential-algebraic equations of first-order into a sequence of first-order linear systems of ordinary differential equations and linear algebraic systems of lower sizes with some necessary conditions on the right-hand sides. In this section, we will review this algorithm by adapting it to matrix differential operators of first-order with formal power series coefficients of the form

$$L = A(x)\,\partial + B(x), \tag{5.3}$$

where $A(x)$, $B(x) \in \mathbb{K}[[x]]^{m\times n}$. Following the steps detailed in [52], we will construct two unimodular matrix differential operators S and T such that the output operator $S\,L\,T$ is, up to a permutation, a block diagonal matrix differential operator having at most a diagonal block of the form

$$\begin{pmatrix} x^p I_q\,\partial + D(x) & 0 \\ 0 & 0 \end{pmatrix}, \tag{5.4}$$

where $p \in \mathbb{N}$, $q \in \mathbb{N}^*$ and $D(x)$ is a square matrix with formal power series coefficients. Its other diagonal blocks are of the form

$$\begin{pmatrix} 0 & C(x) \\ 0 & 0 \end{pmatrix}, \tag{5.5}$$

where $C(x)$ is an invertible matrix with formal power series coefficients.

Remark 5.2.1. The matrix in (5.4), respectively in (5.5), may of course have one of the following forms:

$$x^p I_q\,\partial + D(x), \quad \begin{pmatrix} x^p I_q\,\partial + D(x) & 0 \end{pmatrix}, \quad \text{or} \quad \begin{pmatrix} x^p I_q\,\partial + D(x) \\ 0 \end{pmatrix},$$
$$\text{respectively } C(x), \quad \begin{pmatrix} C(x) \\ 0 \end{pmatrix}, \quad \text{or} \quad \begin{pmatrix} 0 & C(x) \end{pmatrix}.$$

The aim of our presentation is twofold: on the one hand, we would like to raise awareness of this algorithm, as we have not found references to it within the Computer Algebra community. On the other hand, our description of this algorithm as a series of reversible transformations on the input operator L given by (5.3) makes easier to understand the method. This presentation also makes it particularly suitable for an implementation in a Computer Algebra system.

The following lemma constitutes an essential step in Harris et al. algorithm.

Lemma 5.2.1. *Let A be a matrix of $\mathbb{K}[[x]]^{m\times n}$. It is always possible to construct two invertible matrices $S \in \mathbb{K}[[x]]^{m\times m}$ and $T \in \mathbb{K}[[x]]^{n\times n}$ such that $S\,A\,T$ has one of the forms*

$$\begin{pmatrix} \widetilde{A} & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & \widetilde{A} \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ \widetilde{A} & 0 \end{pmatrix}, \quad \textit{or} \quad \begin{pmatrix} 0 & 0 \\ 0 & \widetilde{A} \end{pmatrix}, \tag{5.6}$$

where $\widetilde{A} \in \mathbb{K}[[x]]^{r\times r}$ is an invertible matrix with r being the rank of A.

Proof. The proof is similar to that of [52, Lemma 1]. □

Note that the matrices S and T in Lemma 5.2.1 may be chosen so that $\widetilde{A} = x^k I_r$ with $k \in \mathbb{N}$.

In what follows, we will present the different steps of the algorithm of [52]. For ease of presentation, we will continue using the same symbols L, A, B, m, n etc, for the different steps whenever no confusion arises. We will denote by $r(L)$ the rank of the leading coefficient matrix of the operator L, *i.e.*, for L given by (5.3), $r(L)$ is by definition equal to $\mathrm{rank}(A)$.

5.2.1 Step 1: normalization

The first step is a normalization step achieved by applying Lemma 5.2.1 to the leading coefficient matrix A of the operator L given by (5.3). Let $S \in \mathbb{K}[[x]]^{m\times m}$ and $T \in \mathbb{K}[[x]]^{n\times n}$ be such that $S\,A\,T = \mathrm{diag}\big(x^k I_r, 0\big)$ where $k \in \mathbb{N}$ and $r = \mathrm{rank}(A) \leq \min\{m, n\}$. Thus, L is equivalent to the operator

$$S\,L\,T = \begin{pmatrix} x^k I_r\,\partial + B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \tag{5.7}$$

where $B_{11} \in \mathbb{K}[[x]]^{r\times r}$ and $B_{22} \in \mathbb{K}[[x]]^{(m-r)\times(n-r)}$. We will refer to the form given by the right-hand side of (5.7) as a *normalized form*. After computing a normalized form, the algorithm proceeds to Step 2.

5.2.2 Step 2: algebraic reduction

Assume that L is in normalized form as in the right-hand side of (5.7). If $B_{22} = 0$, we go to Step 3. Otherwise, we apply Lemma 5.2.1 to B_{22} and obtain two matrices $S \in \mathbb{K}[[x]]^{(m-r)\times(m-r)}$ and $T \in \mathbb{K}[[x]]^{(n-r)\times(n-r)}$ such that $S\,B_{22}\,T = \mathrm{diag}\Big(0, \widetilde{B}_{33}\Big)$ with $\widetilde{B}_{33}$ an invertible matrix of size a ($0 < a \leq \min\{m-r, n-r\}$). Thus, multiplying L on the left and the right respectively by $\mathrm{diag}(I_r, S)$ and $\mathrm{diag}(I_r, T)$, we find

$$\widetilde{L} = \begin{pmatrix} x^k I_r\,\partial + B_{11} & \widetilde{B}_{12} & \widetilde{B}_{13} \\ \widetilde{B}_{21} & 0 & 0 \\ \widetilde{B}_{31} & 0 & \widetilde{B}_{33} \end{pmatrix} \in \mathbb{K}[[x]][\partial]^{m\times n}. \tag{5.8}$$

The latter operator can be further simplified using the transformations

$$S_1 = \begin{pmatrix} I_r & 0 & -\widetilde{B}_{13}\,\widetilde{B}_{33}^{-1} \\ 0 & I_{m-r-a} & 0 \\ 0 & 0 & I_a \end{pmatrix} \text{ on the left and } T_1 = \begin{pmatrix} I_r & 0 & 0 \\ 0 & I_{n-r-a} & 0 \\ -\widetilde{B}_{33}^{-1}\,\widetilde{B}_{31} & 0 & I_a \end{pmatrix} \text{ on the right.}$$

This allows to eliminate $\widetilde{B}_{13}$ and $\widetilde{B}_{31}$ in (5.8). Thus, we get an operator of the form

$$\overline{L} = S_1\,\widetilde{L}\,T_1 = \begin{pmatrix} x^k\,I_r\,\partial + B_{11} - \widetilde{B}_{13}\,\widetilde{B}_{33}^{-1}\,\widetilde{B}_{31} & \widetilde{B}_{12} & 0 \\ \widetilde{B}_{21} & 0 & 0 \\ 0 & 0 & \widetilde{B}_{33} \end{pmatrix}.$$

Let $\alpha = \min\Big\{v\Big(B_{11} - \widetilde{B}_{13}\,\widetilde{B}_{33}^{-1}\,\widetilde{B}_{31}\Big), v\Big(\widetilde{B}_{12}\Big), k\Big\}$ and multiply $\overline{L}$ on the left by $\mathrm{diag}(x^{-\alpha} I_r, I_{m-r})$. We find

$$\left(\begin{array}{cc|c} x^q I_r\,\partial + \widehat{B}_{11} & \widehat{B}_{12} & 0 \\ \widetilde{B}_{21} & 0 & 0 \\ \hline 0 & 0 & \widetilde{B}_{33} \end{array}\right) = \left(\begin{array}{c|c} L_{11} & 0 \\ \hline 0 & \widetilde{B}_{33} \end{array}\right) \tag{5.9}$$

where $q = k - \alpha$, $\widehat{B}_{11} = x^{-\alpha}\left(B_{11} - \widetilde{B}_{13}\,\widetilde{B}_{33}^{-1}\,\widetilde{B}_{31}\right)$ and $\widehat{B}_{12} = x^{-\alpha}\,\widetilde{B}_{12}$. The operator to consider now is $L_{11} \in \mathbb{K}[[x]][\partial]^{(m-a)\times(n-a)}$ which is in normalized form (5.7) with $B_{22} = 0$ and hence we proceed to Step 3. Note that here we have simplified the presentation of [52] by combining Steps I (ii) and I (iii) in [52, Section 3].

5.2.3 Step 3: differential row-reduction

We can now assume that L is of the form

$$L = \begin{pmatrix} x^q I_r\,\partial + B_{11} & B_{12} \\ B_{21} & 0 \end{pmatrix} \in \mathbb{K}[[x]][\partial]^{m\times n}. \tag{5.10}$$

If $B_{21} = 0$, then we go to Step 4. Otherwise, we compute two matrices $S \in \mathbb{K}[[x]]^{(m-r)\times(m-r)}$ and $T \in \mathbb{K}[[x]]^{r\times r}$ as in Lemma 5.2.1 such that $S\,B_{21}\,T = \operatorname{diag}\left(\widetilde{B}_{31}, 0\right)$ where $\widetilde{B}_{31}$ is an invertible matrix of size s $(0 < s \leq r)$. Then, put

$$\overline{L} = \operatorname{diag}\left(T^{-1}, S\right) L \operatorname{diag}(T, I_{n-r}) = \begin{pmatrix} x^q\,I_r\,\partial + x^q\,T^{-1}T' + T^{-1}B_{11}\,T & T^{-1}B_{12} \\ S\,B_{21}\,T & 0 \end{pmatrix},$$

where $T' = \frac{dT}{dx}$. Let $\beta = \min\left\{v\left(x^q\,T^{-1}T' + T^{-1}B_{11}T\right), v\left(T^{-1}B_{12}\right), q\right\}$ and multiply $\overline{L}$ on the left by $\operatorname{diag}\left(x^{-\beta}I_r, I_{m-r}\right)$. Partition the resulting operator into blocks compatible with the partition of $\operatorname{diag}\left(\widetilde{B}_{31}, 0\right)$, *i.e.*,

$$\widetilde{L} = \operatorname{diag}\left(x^{-\beta}\,I_r, I_{m-r}\right)\overline{L} = \begin{pmatrix} x^{q-\beta}\,I_s\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12} & \widetilde{B}_{13} \\ \widetilde{B}_{21} & x^{q-\beta}\,I_{r-s}\,\partial + \widetilde{B}_{22} & \widetilde{B}_{23} \\ \widetilde{B}_{31} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Now, since $\widetilde{B}_{31}$ is invertible, then by left-multiplication, we can eliminate all the blocks above $\widetilde{B}_{31}$. This can be made by multiplying $\widetilde{L}$ on the left by

$$\widetilde{S} = \begin{pmatrix} 0 & I_{r-s} & -\widetilde{B}_{21}\widetilde{B}_{31}^{-1} & 0 \\ I_s & 0 & -x^{q-\beta}\left((\widetilde{B}_{31}^{-1})' + \widetilde{B}_{31}^{-1}\,\partial\right) - \widetilde{B}_{11}\widetilde{B}_{31}^{-1} & 0 \\ 0 & 0 & I_s & 0 \\ 0 & 0 & 0 & I_{m-r-s} \end{pmatrix}$$

and we obtain

$$\left(\begin{array}{c|cc} 0 & x^{q-\beta}I_{r-s}\,\partial + \widetilde{B}_{22} & \widetilde{B}_{23} \\ 0 & \widetilde{B}_{12} & \widetilde{B}_{13} \\ \hline \widetilde{B}_{31} & 0 & 0 \\ 0 & 0 & 0 \end{array}\right) = \left(\begin{array}{c|c} 0 & L_{12} \\ \hline \widetilde{B}_{31} & 0 \\ 0 & 0 \end{array}\right).$$

The operator to consider now is L_{12} which is in normalized form of smaller size than the initial operator L given by (5.10) and satisfies $r(L_{12}) < r(L)$. The algorithm now proceeds recursively to Step 2.

Thus, after successive applications of Steps 2 and 3, we either find an operator of the form (5.10) for which $B_{21} = 0$, in which case, we proceed to Step 4 or an operator of order zero in which case the algorithm terminates.

5.2.4 Step 4: differential column-reduction

We can assume now that we have an operator of the form

$$L = \begin{pmatrix} x^p I_r\,\partial + B_{11} & B_{12} \\ 0 & 0 \end{pmatrix}. \tag{5.11}$$

If $B_{12} = 0$, then the algorithm is completed. Otherwise, let S, T such that $SB_{12}T = \operatorname{diag}\big(0, \overline{B}_{24}\big)$ where $\overline{B}_{24}$ is an invertible matrix of size $b > 0$. Put $\overline{L} = \operatorname{diag}(S, I_{m-r})\, L \operatorname{diag}\big(S^{-1}, T\big)$ and partition $\overline{L}$ into blocks compatible with the partition of $\operatorname{diag}\big(0, \overline{B}_{24}\big)$, *i.e.*,

$$\begin{aligned}
\overline{L} &= \operatorname{diag}(S, I_{m-r})\, L \operatorname{diag}\big(S^{-1}, T\big) \\
&= \begin{pmatrix} x^p\, I_r\,\partial + x^p\, S\,(S^{-1})' + S\, B_{11}\, S^{-1} & S\, B_{12}\, T \\ 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} x^p\, I_{r-b}\,\partial + \overline{B}_{11} & \overline{B}_{12} & 0 & 0 \\ \overline{B}_{21} & x^p\, I_b\,\partial + \overline{B}_{22} & 0 & \overline{B}_{24} \\ 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{K}((x))[\partial]^{m\times n}.
\end{aligned}$$

Now, multiply the first r rows of $\overline{L}$ by a suitable power of x to get an operator with coefficients in $\mathbb{K}[[x]]$, *i.e.*,

$$\widetilde{L} = \begin{pmatrix} x^q\, I_{r-b}\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12} & 0 & 0 \\ \widetilde{B}_{21} & x^q\, I_b\,\partial + \widetilde{B}_{22} & 0 & \widetilde{B}_{24} \\ 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{K}[[x]][\partial]^{m\times n},$$

where $q \in \mathbb{N}$ and $\widetilde{B}_{24}$ is an invertible matrix. Let $\widetilde{S}$ be the constant permutation matrix and $\widetilde{T}$ the unimodular matrix differential operator defined as below

$$\widetilde{S} = \begin{pmatrix} I_{r-b} & 0 & 0 \\ 0 & 0 & I_{m-r-b} \\ 0 & I_b & 0 \end{pmatrix} \quad \text{and} \quad \widetilde{T} = \begin{pmatrix} I_{r-b} & 0 & 0 & 0 \\ 0 & I_b & 0 & 0 \\ 0 & 0 & I_{n-r-b} & 0 \\ -\widetilde{B}_{24}^{-1}\,\widetilde{B}_{21} & -\widetilde{B}_{24}^{-1}\big(x^q\, I_b\,\partial + \widetilde{B}_{22}\big) & 0 & I_b \end{pmatrix},$$

then $\widetilde{S}\,\widetilde{L}\,\widetilde{T}$ is of the form

$$\widetilde{S}\,\widetilde{L}\,\widetilde{T} = \left(\begin{array}{cc|cc} x^q I_{r-b}\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \widetilde{B}_{24} \end{array}\right) = \left(\begin{array}{c|cc} L_{11} & 0 & 0 \\ \hline 0 & 0 & \widetilde{B}_{24} \end{array}\right).$$

Therefore, we consider now the operator L_{11} which is of the form (5.11) but of smaller size and $r(L_{11}) < r(L)$. We repeat successively this step until we find an operator of the form (5.11) for which B_{12} is either a zero matrix and hence the algorithm is completed, or an invertible matrix in which case, after a right multiplication by a suitable unimodular matrix differential operator, we get an operator of order zero upon which the algorithm terminates.

5.3 A new reduction algorithm

Inspired by the work of Harris et al. [52], we will develop in this section a new reduction algorithm which computes an operator of the form (5.2), equivalent to $L = A(x)\,\partial + B(x) \in \mathbb{K}[[x]][\partial]^{m\times n}$. This algorithm organizes the steps of the algorithm in [52] in two main stages: treating the rows of L (only left-multiplications are authorized), and treating its columns (only right-multiplications

are authorized). Our approach as presented below is simpler than that of Harris et al. and uses a weaker version of Lemma 5.2.1. It is essentially based on the computation of left and right nullspaces of rectangular matrices which makes it suitable for a generalization to systems of higher-order as we will see in the next chapter.

Our algorithm consists in applying alternatively row-reduction and column-reduction as described below until we find an operator equivalent to L which satisfies some properties allowing the decoupling.

5.3.1 Row-reduction

Multiplying operator $L = A\,\partial + B$ on the left by a suitable matrix of $\mathrm{GL}_m(\mathbb{K}((x)))$, we can assume that the leading coefficient matrix A is in the block partition

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & 0 \end{pmatrix},$$

where $A_{11} \in \mathbb{K}[[x]]^{r\times r}$ and $r = r(L) = \mathrm{rank}(A)$. Write $B = (B_{ij})_{i,j=1,2}$ decomposed into blocks as matrix A. The operator to consider is then of the form

$$L = A\,\partial + B = \begin{pmatrix} A_{11}\,\partial + B_{11} & A_{12}\,\partial + B_{12} \\ B_{21} & B_{22} \end{pmatrix}. \tag{5.12}$$

Lemma 5.3.1. *Given a matrix differential operator L of the form (5.12), assume that*

$$\mathrm{rank}\begin{pmatrix} A_{11} & A_{12} \\ B_{21} & B_{22} \end{pmatrix} < \mathrm{rank}\begin{pmatrix} A_{11} & A_{12} \end{pmatrix} + \mathrm{rank}\begin{pmatrix} B_{21} & B_{22} \end{pmatrix}. \tag{5.13}$$

Then, there exists a unimodular matrix differential operator S such that $r(SL) < r(L)$.

Proof. Equation (5.13) is equivalent to saying that there exists $i \in \{1,\dots,r\}$ such that

$$A(i,.) = \sum_{\substack{k=1\\k\neq i}}^{r} \alpha_k\, A(k,.) + \sum_{j=1}^{m-r} \beta_j\, B(r+j,.),$$

where $\alpha_k, \beta_j \in \mathbb{K}((x))$ and at least one of the β_j is nonzero (since the nonzero rows of A are supposed to be linearly independent). Let S be the matrix differential operator obtained from the identity matrix of size m by replacing the ith row by

$$\begin{pmatrix} -\alpha_1 & \cdots & -\alpha_{i-1} & 1 & -\alpha_{i+1} & \cdots & -\alpha_r & -\beta_1\,\partial & \cdots & -\beta_{m-r}\,\partial \end{pmatrix}.$$

Note that S is unimodular since a left-multiplication by S consists in performing a series of elementary row operations. Multiplying L on the left by S leaves the rows of index $k \neq i$ unchanged and replaces the ith row by

$$A(i,.)\,\partial + B(i,.) - \sum_{\substack{k=1\\k\neq i}}^{r} \alpha_k\,(A(k,.)\,\partial + B(k,.)) - \sum_{j=1}^{m-r} \beta_j\,(\partial(B(r+j,.)) + B(r+j,.)\,\partial) =$$

$$B(i,.) - \sum_{\substack{k=1\\k\neq i}}^{r} \alpha_k\, B(k,.) - \sum_{j=1}^{m-r} \beta_j\,\partial(B(r+j,.)),$$

as if we replace the ith row of A by a zero row and that of B by $B(i,.) - \sum_{\substack{k=1\\k\neq i}}^{r} \alpha_k\, B(k,.) - \sum_{j=1}^{m-r} \beta_j\,\partial(B(r+j,.))$. Hence, we have $r(SL) < r(L)$. □

We repeat successively Lemma 5.3.1 until we find an operator equivalent to L of the form

$$\widetilde{L} = \widetilde{A}\,\partial + \widetilde{B} = \begin{pmatrix} \widetilde{A}_{11}\,\partial + \widetilde{B}_{11} & \widetilde{A}_{12}\,\partial + \widetilde{B}_{12} \\ \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix}$$

with $\begin{pmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \end{pmatrix}$ of size $q \times n$, where $q = r(\widetilde{L}) \leq r(L)$ (*i.e.*, it has full row-rank), and

$$\operatorname{rank} \begin{pmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \\ \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix} = \operatorname{rank} \begin{pmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \end{pmatrix} + \operatorname{rank} \begin{pmatrix} \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix}. \tag{5.14}$$

The final sub-step of the row-reduction consists in multiplying $\widetilde{L}$ on the left by a convenient matrix of $\mathrm{GL}_m(\mathbb{K}((x)))$ to eliminate all the linearly dependent rows in $\begin{pmatrix} \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix}$. The resulting operator is said to be *row-reduced*.

Example 5.3.1. *Consider the matrix differential operator of first-order*

$$L = A(x)\,\partial + B(x) = \begin{pmatrix} 1 & 1-x & 2-x & 0 & -1 \\ 0 & 1 & 0 & 0 & 1 \\ x & 1 & 1 & 0 & x \\ 1+x & 1-x & 3-x & 0 & x-2 \\ 2+2\,x & 4-2\,x & 6-2\,x & 0 & 2\,x-2 \\ 1 & -x & 2-x & 0 & -2 \end{pmatrix} \partial + \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & -2 \\ 1 & x & 2+x & 2 & -2 \\ 1 & 1 & 3 & 0 & -2 \end{pmatrix}. \tag{5.15}$$

A basis of the left nullspace of A is given by the rows of the matrix

$$\begin{pmatrix} -1 & 1 & -1 & 1 & 0 & 0 \\ -2 & 0 & -2 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Thus, multiplying operator L on the left by the invertible matrix

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 \\ -2 & 0 & -2 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

yields the operator

$$\widehat{L} = \widehat{A}(x)\,\partial + \widehat{B}(x) = \left(\begin{array}{ccc|cc} 1 & 1-x & 2-x & 0 & -1 \\ 0 & 1 & 0 & 0 & 1 \\ x & 1 & 1 & 0 & x \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) \partial + \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 2 & 0 & -2 \\ 1 & x & 2+x & 0 & -2 \\ 1 & 1 & 3 & 0 & -2 \end{array}\right)$$

to which Lemma 5.3.1 can be applied (here $r(\widehat{L}) = 3$). Indeed, the following relation

$$\widehat{A}(1,.) = \widehat{A}(2,.) + 2\,\widehat{B}(4,.) - \widehat{B}(5,.)$$

is satisfied. Thus, the leading coefficient matrix of the operator obtained by multiplying $\widehat{L}$ *on the left by the unimodular matrix differential operator*

$$S_1 = \begin{pmatrix} 1 & -1 & 0 & -2\,\partial & \partial & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

will have its first row equal to zero. Permuting the first three rows, $S_1\,\widehat{L}$ *is equivalent to*

$$\widetilde{L} = \widetilde{A}(x)\,\partial + \widetilde{B}(x) = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & 0 & 1 \\ x & 1 & 1 & 0 & x \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)\partial + \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 & -2 \\ 1 & x & 2+x & 0 & -2 \\ 1 & 1 & 3 & 0 & -2 \end{array}\right)$$

which satisfies the condition (5.14). Here, we have decreased the value of $r(L)$ *by one since* $r(\widetilde{L}) = 2 < r(L) = 3$. *The last step now consists in multiplying* $\widetilde{L}$ *on the left by*

$$S_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -x & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 \end{pmatrix}$$

to eliminate the linearly dependent rows in the second block row of $\widetilde{B}(x)$. *We finally obtain the row-reduced operator*

$$\overline{L} = S_2\,\widetilde{L} = \overline{A}(x)\,\partial + \overline{B}(x) = \left(\begin{array}{ccc|cc} 0 & 1 & 0 & 0 & 1 \\ x & 1 & 1 & 0 & x \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)\partial + \left(\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right). \tag{5.16}$$

■

5.3.2 Column-reduction

Assume now that the matrix differential operator $L = A\,\partial + B$ is in the form (5.12) with $\begin{pmatrix} A_{11} & A_{12} \end{pmatrix}$ of size $q \times n$ and $r(L) = q$. Let T_1 be an invertible matrix such that

$$\widetilde{A} = A\,T_1 = \begin{pmatrix} \widetilde{A}_{11} & 0 \\ 0 & 0 \end{pmatrix}$$

with $\widetilde{A}_{11}$ a $q \times q$ invertible matrix. Denote $A\,T_1' + B\,T_1$ by $\widetilde{B} = \begin{pmatrix} \widetilde{B}_{ij} \end{pmatrix}_{i,j=1,2}$. Hence, L is equivalent to

$$\widetilde{L} = L\,T_1 = \begin{pmatrix} \widetilde{A}_{11}\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12} \\ \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix}. \tag{5.17}$$

The following lemma can be proved similarly to Lemma 5.3.1.

Lemma 5.3.2. *Given a matrix differential operator $\widetilde{L}$ of the form (5.17), we assume that*

$$\operatorname{rank}\begin{pmatrix}\widetilde{A}_{11} & \widetilde{B}_{12}\\ 0 & \widetilde{B}_{22}\end{pmatrix} < \operatorname{rank}\left(\widetilde{A}_{11}\right) + \operatorname{rank}\begin{pmatrix}\widetilde{B}_{12}\\ \widetilde{B}_{22}\end{pmatrix}.$$

Then, there exists a unimodular matrix differential operator T_2 such that $r\left(\widetilde{L}\,T_2\right) < r\left(\widetilde{L}\right)$.

Applying successively Lemma 5.3.2, we find an equivalent operator

$$\overline{L} = \overline{A}\,\partial + \overline{B} = \begin{pmatrix}\overline{A}_{11}\,\partial + \overline{B}_{11} & \overline{B}_{12}\\ \overline{B}_{21} & \overline{B}_{22}\end{pmatrix}$$

where $\overline{A}_{11}$ is of size $q \times r(\overline{L})$ $\left(r(\overline{L}) \leq q = r\left(\widetilde{L}\right)\right)$ satisfying

$$\operatorname{rank}\begin{pmatrix}\overline{A}_{11} & \overline{B}_{12}\\ 0 & \overline{B}_{22}\end{pmatrix} = \operatorname{rank}\left(\overline{A}_{11}\right) + \operatorname{rank}\begin{pmatrix}\overline{B}_{12}\\ \overline{B}_{22}\end{pmatrix} \tag{5.18}$$

Finally, we multiply $\overline{L}$ on the right by a matrix of $\mathrm{GL}_n(\mathbb{K}((x)))$ to cancel all linearly dependent columns of $\begin{pmatrix}\overline{B}^{12}\\ \overline{B}^{22}\end{pmatrix}$. The resulting operator is said to be *column-reduced*.

Example 5.3.2. *Consider the first-order matrix differential operator $\overline{L}$ given by (5.16). Working with the columns as explained above, one can obtain the unimodular matrix differential operator*

$$T = \begin{pmatrix}1 & -1 & -1 & 0 & -2\,x\\ 0 & x & 0 & 0 & -3\,x\\ 0 & 0 & x & 0 & 3\,x\\ 0 & -x\,\partial & 0 & 1 & x\,\partial + 1\\ 0 & 0 & 0 & 0 & 2\,x\end{pmatrix}$$

so that

$$\overline{L}\,T = \left(\begin{array}{c|cccc}0 & 0 & 0 & 0 & 0\\ x & 0 & 0 & 0 & 0\\ \hline 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{array}\right)\partial + \left(\begin{array}{c|ccccc}0 & 1 & 0 & 1 & 0\\ 0 & 1 & 1 & 0 & 0\\ \hline 0 & x & x & 0 & 0\\ 1 & -1 & 2\,x-1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{array}\right) \tag{5.19}$$

is column-reduced. ■

5.3.3 Decoupling differential and algebraic equations

As we have seen, each time Lemma 5.3.1 or Lemma 5.3.2 is applied, the rank of the leading coefficient matrix of the operator decreases by one. Hence, after carrying out a finite number of times row- and column-reductions[2], L will be equivalent to matrix differential operator which is simultaneously row and column reduced. The latter operator is a block diagonal one $\operatorname{diag}\left(\widetilde{L}_1, 0\right)$ with $\widetilde{L}_1$ of size $p \times s$ of the form

$$\widetilde{L}_1 = \begin{pmatrix}\widetilde{A}_{11}\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12}\\ \widetilde{B}_{21} & \widetilde{B}_{22}\end{pmatrix}, \tag{5.20}$$

[2]We will see in the next chapter that this is after at most the second application of row-reduction.

where $\widetilde{A}_{11}$ is a $d \times d$ invertible matrix,

$$\operatorname{rank} \begin{pmatrix} \widetilde{A}_{11} & 0 \\ \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix} = p \tag{5.21}$$

and

$$\operatorname{rank} \begin{pmatrix} \widetilde{A}_{11} & \widetilde{B}_{12} \\ 0 & \widetilde{B}_{22} \end{pmatrix} = s. \tag{5.22}$$

Since $\widetilde{A}_{11}$ is invertible, Equation (5.21), respectively Equation (5.22), implies that $\widetilde{B}_{22}$ has full row-rank, respectively full column-rank. Consequently, $\widetilde{B}_{22}$ is invertible and $p = s$.

Now, using the fact that $\widetilde{B}_{22}$ is invertible, it is easy to decouple the operator $\widetilde{L}_1$. Indeed,

$$\begin{pmatrix} I_d & -\widetilde{B}_{12}\,(\widetilde{B}_{22})^{-1} \\ 0 & I_{p-d} \end{pmatrix} \widetilde{L}_1 \begin{pmatrix} I_d & 0 \\ -(\widetilde{B}_{22})^{-1}\widetilde{B}_{21} & I_{p-d} \end{pmatrix} = \begin{pmatrix} \widetilde{A}_{11}\,\partial + \widetilde{B}_{11} - \widetilde{B}_{12}\,(\widetilde{B}_{22})^{-1}\widetilde{B}_{21} & 0 \\ 0 & \widetilde{B}_{22} \end{pmatrix}.$$

Multiplying the latter operator by a suitable power of x, we can assume that its coefficients belong to $\mathbb{K}[[x]]$. This leads to the following theorem.

Theorem 5.3.1. *Given a linear DAE of the form (5.1), there exist two unimodular matrix differential operators $S \in \mathbb{K}((x))[\partial]^{m\times m}$ and $T \in \mathbb{K}((x))[\partial]^{n\times n}$ that transform (5.1) into a decoupled system of the form*

$$\widetilde{L}(z) = \begin{pmatrix} \widetilde{A}_{11}\,\partial + \widetilde{B}_{11} & 0 & 0 \\ 0 & \widetilde{B}_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \widetilde{f}_1 \\ \widetilde{f}_2 \\ \widetilde{f}_3 \end{pmatrix}, \tag{5.23}$$

where $\widetilde{L} = S\,L\,T \in \mathbb{K}[[x]][\partial]^{m\times n}$, $\widetilde{A}_{11}$ and $\widetilde{B}_{22}$ are both invertible, $y = T(z)$ and $\widetilde{f} = S(f)$.

If $\widetilde{f}_3 = 0$ in (5.23), then the system (5.23), respectively System (5.1), is consistent. If moreover $\operatorname{rank}(\widetilde{A}_{11}) + \operatorname{rank}(\widetilde{B}_{22}) < n$, then the component z_3 can be chosen as an arbitrary vector function and hence the dimension of the affine solution space of (5.23) is infinite.

Example 5.3.3. *We continue our reduction process on the operator L given by (5.15). Operator (5.19) is equivalent, up to a permutation, to*

$$\left(\begin{array}{c|cccc} x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) \partial + \left(\begin{array}{c|ccc|c} 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 0 & x & x & 0 & 0 \\ 1 & -1 & 2\,x-1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

which is also row-reduced. Thus, multiplying the above operator on the left and on the right respectively by

$$\begin{pmatrix} 1 & 0 & -\frac{1}{x} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2\,x} & 1 & 0 & 0 & 0 \\ \frac{-1}{2\,x} & 0 & 1 & 0 & 0 \\ \frac{-1}{2\,x} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

we end up with the decoupled operator

$$L_f = \left(\begin{array}{c|cccc} x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right) \partial + \left(\begin{array}{c|ccc|c} 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & -1 & 2\,x-1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Hence, solving system $L(y) = f$ is reduced to solving a scalar linear differential equation and a linear algebraic system of size 3. *Finally, we note that the two operators L_f and L are connected by $L_f = S_f\, L\, T_f$ with S_f and T_f two unimodular matrix differential operators respectively given by*

$$S_f = \begin{pmatrix} -\frac{1}{x} & \frac{2\,\partial+1}{x} & 1 & 2\,\frac{\partial}{x} & -\frac{\partial}{x} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2\,\partial-1 & 0 & -2\,\partial & \partial & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 \\ -1-x & 2\,x\,\partial-1+x & -1 & 2\,x\,\partial-1 & 1-x\,\partial & 0 \\ -1 & 2\,\partial+1 & 1 & 2\,\partial-1 & -\partial & 1 \end{pmatrix} \text{ and } T_f = \begin{pmatrix} 1 & -1 & -1 & 0 & -2\,x \\ \frac{1}{2} & x & 0 & 0 & -3\,x \\ -\frac{1}{2} & 0 & x & 0 & 3\,x \\ \frac{1}{2}\,\partial & -x\,\partial & 0 & 1 & x\,\partial+1 \\ 0 & 0 & 0 & 0 & 2\,x \end{pmatrix}.$$

■

Example 5.3.4. *We consider the linear DAE*

$$L(y) = \begin{pmatrix} 1 & x & 1 & -x \\ x^2 & 2+x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} y'(x) + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1+x & -x \\ 0 & 0 & -1 & x \end{pmatrix} y(x) = f(x),$$

where $f(x)$ is a 3*-dimensional arbitrary vector. Applying our algorithm to L, we find*

$$S = \begin{pmatrix} -x^2 & 1 & \frac{-7x-2x^2-2+x^3}{-2-x+x^3} \\ -2-x+x^3 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} -x & \frac{2+x}{-2-x+x^3} & 0 & -1 \\ 1 & \frac{-x^2}{-2-x+x^3} & 0 & 0 \\ x & 1 & x & x\,\partial \\ 1 & 0 & 1 & \partial \end{pmatrix}$$

and the decoupled operator $\widetilde{L} = S\, L\, T$ is given by

$$\widetilde{L} = \begin{pmatrix} (-x^3+x+2)\,\partial & 0 & 0 & 0 \\ 0 & x^2 & x^3-x-2 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}.$$

Here, the system is consistent for any right-hand side $f(x)$. ■

5.3.4 Application: classification of singularities.

Our reduction algorithm makes possible to extend the classification of the singularity at $x = 0$ (as regular or singular singularity) to linear DAEs since it reduces the problem to the ODE case.

Definition 5.3.1. *Given an homogeneous linear DAE of the form (5.1), we say that the point $x = 0$ is a* regular singularity, *if it is a regular singularity of $\widetilde{A}_{11}(x)\, z_1'(x) + \widetilde{B}_{11}(x)\, z_1(x) = 0$ and* $\operatorname{rank}(\widetilde{A}_{11}) + \operatorname{rank}(\widetilde{B}_{22}) = n$, *where $\widetilde{A}_{11}$, $\widetilde{B}_{11}$ and $\widetilde{B}_{22}$ are given by (5.23). Otherwise, we say that it is an* irregular singularity.

Thus, by applying our reduction algorithm and using techniques developed for the ODEs (*e.g.*, using Moser's algorithm [77] on the system $z_1'(x) + \widetilde{A}_{11}^{-1}(x)\, \widetilde{B}_{11}(x)\, z_1(x) = 0$), we are able to algorithmically decide whether a given homogeneous linear DAE has a regular or irregular singularity at $x = 0$. However, it is currently an open problem how to algorithmically classify singularities without this decoupling.

Chapter 6

On Simultaneous Row and Column Reduction of Higher-Order Linear Differential Systems

Contents

6.1 Introduction

We consider linear differential systems of the form

$$L(y(x)) = \sum_{i=0}^{\ell} A_i(x)\, y^{(i)}(x) = f(x), \tag{6.1}$$

where x is a complex variable, $A_i(x) \in \mathbb{K}[[x]]^{m\times n}$ and $f(x) \in \mathbb{K}[[x]]^m$. Such systems arise naturally in many applications such as multi-body systems, models of electrical circuits, robotic modelling and mechanical systems (see [73, 82, 89] and references therein). A standard technique for dealing with systems of the form (6.1) is to transform them to a system having the same properties but which is in a "simpler" form where, for instance, a local analysis is more readily determined.

One way for obtaining systems equivalent to (6.1) is by performing reversible row operations on the matrix differential operator

$$L = \sum_{i=0}^{\ell} A_i(x)\, \partial^i.$$

As an example, the differential row Hermite normal form [49] results in a system having a triangular form and hence it can useful for solving such systems. However, the differential equations obtained are often of order higher than that of the original system. A second form, the differential row Popov normal form [40], always yields a matrix differential operator of order at most that of the original one. The utility of this form is that it can easily be transformed into a first-order differential-algebraic equation. One can also obtain equivalent systems making

use of both reversible row and column operations on L. The most extreme example of this is the transformation of the matrix differential operator L to a diagonal form, called the Jacobson normal form [70, 74]. The latter form reduces System (6.1) into a scalar linear differential equation with, often, very large coefficients especially when the system is of large size. Hence, from an algorithmic viewpoint, it is better to manipulate systems directly rather than to convert them into scalar differential equations.

In this chapter, we give two algorithms for reducing systems (6.1) into a new form called *simultaneously row and column reduced form* The first algorithm follows the techniques used in Section 5.3 of the previous chapter while the second algorithm makes use of reduction techniques used for matrix polynomial [28, 27, 78]. Both algorithms reduce orders in rows and columns by carrying out of a series of elementary (block) row and column operations. These methods are also extended to handle *two-sided block Popov forms* (see Definition 6.4.2), a special form of matrix of differential operators which generalizes the Popov normal form. We give a complexity analysis of our algorithms and also illustrate their use in the special case where the operator is of first-order.

While we have chosen to focus our work on matrix differential operators with coefficients in the domain of formal power series $\mathbb{K}[[x]]$, our methods easily extend to the field of rational functions $\mathbb{K}(x)$. In fact, our algorithms are entirely algebraic involving only basic row and column operations. Everything that we do can be done for the more general case of Ore matrix polynomials having coefficients in the field of rational functions $\mathbb{K}(x)$.

The rest of the chapter is divided as follows. In Section 6.2, we review the notion of row-reduction. In Section 6.3, we detail the concept of a simultaneously row and column reduced matrix differential operator and we give a first algorithm for its computation. This algorithm extends the one for first-order matrix differential operators presented in Section 5.3 of the previous chapter. In Section 6.4, we develop a second algorithm for the construction of a simultaneously row and column reduced form. We also introduce the notion of a two-sided block Popov form for a matrix differential operator and propose a procedure for its computation. In Section 6.5, we give an algorithmic description and provide a complexity analysis for the procedures developed in the previous section, while in Section 6.6, we discuss the reduction of higher-order differential systems into a purely algebraic part and a purely differential part.

This chapter is the subject of the paper [19].

Notation. For a nonzero matrix differential operator L, we denote by $\ell c(L)$ its leading coefficient matrix and by $\mathrm{ord}(L)$ its order. When $L = 0$, we set $\mathrm{ord}(L) = -\infty$ and $\ell c(L) = 0$. If L can be partitioned into blocks L_{ij} for $1 \leq i, j \leq k$, then we denote by $L_{i,*}$ the ith block row of L, *i.e.*, $L_{i,*} = \begin{pmatrix} L_{i1} & L_{i2} & \cdots & L_{ik} \end{pmatrix}$, and by $L_{*,j}$ the jth block column of L. For $\delta = (\delta_1, \ldots, \delta_n) \in (\mathbb{N} \cup \{\pm\infty\})^n \backslash \{\pm\infty\}^n$, we denote by $|\delta|$ the following sum $\sum_{1 \leq i \leq n \text{ s.t. } \delta_i \neq \pm\infty} \delta_i$.

6.2 Row-reduction procedure

In the case of a scalar differential operator, the leading coefficient, *i.e.*, the coefficient of the highest-order term, plays an important role in a number of tasks, as for example in singularity analysis and finding local solutions. In the case of a matrix differential operator, this is more complicated by the fact that there are many ways to define a leading coefficient, *e.g.*, the matrix of highest-order term, the matrix of highest row-order terms or the matrix of highest column-order terms. Invertibility conditions are then important. A matrix differential operator having an invertible leading row coefficient matrix is said to be in a *row-reduced form* [65] (a *column-reduced form* for the corresponding notion on columns). More precisely,

Definition 6.2.1. *Let* $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ *and let* $\delta_i = \operatorname{ord}(L(i.,))$ *for* $i = 1, \ldots, m$.

1. $\delta = (\delta_1, \ldots, \delta_m)$ *is called the* row-order *of* L.
2. *The* leading row coefficient matrix *of* L *is the* $m \times n$ *matrix with the* (i, j) *entry being the coefficient of order* δ_i *of the* (i, j) *entry of* L.
3. L *is said to be* row-reduced *if the nonzero rows of its leading row coefficient matrix are linearly independent over* $\mathbb{K}((x))$.

For computing a row-reduced form, we can make use of a procedure developed by Beckermann and Labahn [28] for the commutative case of matrix polynomials (see also [65, page 386]) and later generalized for Ore matrix polynomials in [27]. In this section, we will review this method for use with matrix differential operators.

Lemma 6.2.1 ([27], Appendix)**.** *The rank*[1] *of a row-reduced matrix differential operator is equal to the rank of its leading row coefficient matrix.*

The following lemma shows that any matrix differential operator can be transformed into a row-reduced form by means of elementary row operations.

Lemma 6.2.2 ([27], Theorem 2.2)**.** *Let* $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ *be a matrix differential operator of rank* $s \leq \min(m, n)$. *There exists a unimodular matrix differential operator* $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ *such that* UL *is of the form*

$$UL = \begin{pmatrix} L^* \\ 0 \end{pmatrix},$$

where L^* *is a row-reduced matrix differential operator of size* $s \times n$ *such that* $\operatorname{ord}(L^*) \leq \operatorname{ord}(L)$ *and all its rows are nonzero.*

For the sake of completeness, we recall the proof of the lemma here.

Proof. If L is already row-reduced then $U = I_m$ and we are done. Otherwise, we may suppose, without any loss of generality, that L has all its zero rows at the bottom of the matrix. In this case, the leading row coefficient matrix of L is of the form

$$\begin{pmatrix} L_0 \\ 0 \end{pmatrix},$$

where $L_0 \in \mathbb{K}[[x]]^{k\times n}$ denotes the leading row coefficient matrix of the first k rows of L ($k \geq s$). As L is not row-reduced, L_0 is of rank less than k and hence we can find a nonzero row vector $v = \begin{pmatrix} v_1 & v_2 & \cdots & v_k \end{pmatrix} \in \mathbb{K}[[x]]^{1\times k}$ such that $v\,L_0 = 0$. Select an index $\nu \in \{1, \ldots, k\}$ such that $v_\nu \neq 0$ and $\delta_\nu = \max\{\delta_i \text{ s.t. } v_i \neq 0\}$, where $\delta = (\delta_1, \ldots, \delta_m)$ denotes the row-order of L. Define $U_1 = \operatorname{diag}(U_{11}, I_{m-k})$ with

$$U_{11} = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ v_1\,\partial^{\delta_\nu-\delta_1} & \cdots & v_{\nu-1}\,\partial^{\delta_\nu-\delta_{\nu-1}} & v_\nu & v_{\nu+1}\,\partial^{\delta_\nu-\delta_{\nu+1}} & \cdots & v_k\,\partial^{\delta_\nu-\delta_k} \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 \end{pmatrix} \in \mathbb{K}[[x]][\partial]^{k\times k}.$$

[1] The row rank and column rank of a matrix differential operator are equal (cf. [38, Chap. 8, Th. 1.1]).

Matrix U_1 is unimodular and multiplying L on the left by U_1 leaves the rows of index $i \neq \nu$ unchanged and replaces the νth row by

$$\begin{aligned}\sum_{i=1}^{k} v_i\, \partial^{\,\delta_\nu-\delta_i} L(i,.) &= \sum_{i=1}^{k} v_i\, \ell c(L(i,.))\, \partial^{\,\delta_\nu} + \text{ terms of order less than } \delta_\nu \\ &= \underbrace{v\, L_0}_{=0} \partial^{\,\delta_\nu} + \text{ terms of order less than } \delta_\nu.\end{aligned}$$

Thus, the νth row of $U_1 L$ has order less than the one of L. Repeating this process a finite number of times, we get a unimodular matrix of differential operators U such that UL is of the form

$$UL = \begin{pmatrix} L^* \\ 0 \end{pmatrix},$$

where L^* is a row-reduced matrix differential operator of size $m_1 \times n$ ($m_1 \leq k$) having only nonzero rows. It remains to show that $m_1 = s$. According to [27, Lemma A.3], we have $\operatorname{rank}(L) = \operatorname{rank}(UL) = \operatorname{rank}(L^*)$. On one hand, the rank of L^* is equal to that of its leading row coefficient matrix which is m_1 (see Lemma 6.2.1). On the other hand, we have $\operatorname{rank}(L) = s$ and thus $m_1 = s$. □

ALGORITHM **Row-Reduction**

INPUT: $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ a matrix differential operator of row-order $\delta = (\delta_1, \ldots, \delta_m)$.
OUTPUT: A row-reduced matrix differential operator $L' \in \mathbb{K}[[x]][\partial]^{m\times n}$ and a unimodular matrix differential operator $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ such that $L' = UL$.

INITIALIZATION: Let $L' = L$ and define L'_0 as the leading row coefficient matrix of L'.
Let $\delta' = (\delta'_1, \ldots, \delta'_m) = \delta$ and $U = I_m$.

While the nonzero rows of L'_0 are linearly dependent **do**
1. Compute $v = \begin{pmatrix} v_1 & v_2 & \cdots & v_m \end{pmatrix} \in \mathbb{K}[[x]]^{1\times m} \setminus \{0\}$ in the left nullspace of L'_0;
2. Select an index ν such that $v_\nu \neq 0$ and $\delta'_\nu = \max\{\delta'_i\,;\, v_i \neq 0\}$;
3. Let $L'(\nu,.) = \sum_{i=1}^{m} v_i\, \partial^{\,\delta'_\nu-\delta'_i} L'(i,.)$;
4. Let $U(\nu,.) = \sum_{i=1}^{m} v_i\, \partial^{\,\delta'_\nu-\delta'_i} U(i,.)$;
5. Update L'_0 and δ';

end do;
Return L' and U;

Proposition 6.2.1. *Let L be a matrix differential operator of order ℓ and row-order δ. Algorithm* **Row-Reduction** *returns a row-reduced matrix differential operator equivalent to L in at most $O(m\,n\,(|\delta|+m)\,(m+3\,\ell+|\delta|))$ operations in $\mathbb{K}[[x]]$.*

Proof. Computing an element in the left nullspace of the matrix $L'_0 \in \mathbb{K}[[x]]^{m\times n}$ costs at most $O\big(m^2\,n\big)$ operations in $\mathbb{K}[[x]]$. This gives the cost of Step 1. Since L' has order always bounded by ℓ, the cost of Step 3 is then $O(m\,n\,\ell)$ operations in $\mathbb{K}[[x]]$. From [27, Theorem 2.2], we can deduce that the order of the multiplier U is always bounded by $\ell + |\delta|$, and so for $i = 1, \ldots, m$, $\operatorname{ord}\Big(\partial^{\,\delta'_\nu-\delta'_i} U(i,.)\Big) \leq 2\,\ell + |\delta|$. Hence, Step 4 can be done in at most $O(m\,n\,(2\,\ell+|\delta|))$ operations in $\mathbb{K}[[x]]$. Finally, as the **While loop** is repeated at most $|\delta|+m-1$ times, we obtain a row-reduced matrix differential operator equivalent to L using at most $O(m\,n\,(|\delta|+m)\,(m+3\,\ell+|\delta|))$ operations in $\mathbb{K}[[x]]$. □

Analogous definitions and results can also be stated for column-reduction (where now the leading column coefficient matrix of the nonzero columns has full column rank). Thus, it is possible to construct a unimodular matrix differential operator $V \in \mathbb{K}[[x]][\partial]^{n\times n}$ such that LV is column-reduced.

Remark now that if $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ is column-reduced, respectively row-reduced, and if we multiply it on the left, respectively on the right, by an invertible matrix of $\mathbb{K}[[x]]$, it remains column-reduced, respectively row-reduced.

Lemma 6.2.3. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ be a matrix differential operator with L_0 denoting its leading column coefficient matrix. Let $U \in \mathbb{K}[[x]]^{m\times m}$ be an invertible matrix over $\mathbb{K}((x))$. Then, the column-order of L and that of UL are equal and the leading column coefficient matrix of UL is UL_0.*

Proof. Without any loss of generality, we can assume that all the columns of L are nonzero. The jth column of UL is indeed equal to U multiplied by the jth column of L, *i.e.*,

$$(UL)(.,j) = UL(.,j).$$

Let $\gamma_j = \mathrm{ord}(L(.,j)) \geq 0$. Then $L(.,j)$ can be written as

$$L(.,j) = \ell c(L(.,j))\,\partial^{\gamma_j} + \text{terms of lower order},$$

with $\ell c(L(.,j)) \neq 0$ (since all columns of L are assumed to be nonzero). Therefore,

$$(UL)(.,j) = UL(.,j) = U\,\ell c(L(.,j))\,\partial^{\gamma_j} + \text{terms of lower order},$$

with $U\,\ell c(L(.,j)) \neq 0$ since $\ell c(L(.,j)) \neq 0$ and U is invertible. Thus, we have $\mathrm{ord}((UL)(.,j)) = \mathrm{ord}(L(.,j))$ and $\ell c((UL)(.,j)) = U\,\ell c(L(.,j))$ which ends the proof. □

Corollary 6.2.1. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ be a matrix differential operator and $U \in \mathbb{K}[[x]]^{m\times m}$, respectively $V \in \mathbb{K}[[x]]^{n\times n}$, be an invertible matrix over $\mathbb{K}((x))$. If L is column-reduced, respectively row-reduced, then so is UL, respectively LV.*

6.3 Simultaneously row and column reduced forms

In this section, we introduce the concept of simultaneously row and column reduced matrix differential operators and give a first method for transforming an arbitrary matrix differential operator into a matrix having such a property. When applied to problems of the form (6.1), one obtains a useful algebraic structure for system simplification. For example, when L is both simultaneously row and column reduced and of first-order then System (6.1) can be decoupled into a purely algebraic system and a purely differential one. For higher-order systems one can use such a transformation to extract a purely algebraic part (if it exists) with a second component easily transformable into a square system of ordinary differential equations of first-order (see Section 6.6).

6.3.1 Simultaneous row and column reduction

Let L be an $m \times n$ matrix differential operator of order ℓ and rank $s \leq \min(m,n)$. One can imagine that to construct a simultaneously row and column reduced form equivalent to L, it suffices to apply row-reduction to L followed by column-reduction to the resulting row-reduced form. Unfortunately, this is not always true. In general, such a computation requires several successive iterations of row-reduction and column-reduction as the example below shows. Note that when applying row-reduction, respectively column-reduction, to L as in Lemma 6.2.2, the value of $|\delta|$, respectively the value of $|\gamma|$, may decrease but in the same time the value of $|\gamma|$, respectively of $|\delta|$, may increase.

Example 6.3.1. *Consider the matrix differential operator given by*

$$L = \begin{pmatrix} \partial^3 + x & 2\,\partial^2 & x^2 + x \\ \partial^2 & x\,\partial^2 & 2\,x^2 + 1 \\ \partial & x\,\partial & 1 \end{pmatrix} \tag{6.2}$$

with row and column orders $\delta = (3, 2, 1)$ *and* $\gamma = (3, 2, 0)$ *respectively. Operator* L *is not row-reduced since its leading row coefficient matrix*

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & x & 0 \\ 1 & x & 0 \end{pmatrix}$$

is singular. Following the construction in the proof of Lemma 6.2.2, we multiply L *on the left by*

$$U_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\partial \\ 0 & 0 & 1 \end{pmatrix}.$$

This gives rise to the matrix differential operator

$$L^{[1]} = U_1 L = \begin{pmatrix} \partial^3 + x & 2\,\partial^2 & x^2 + x \\ 0 & -\partial & -\partial + 2x^2 + 1 \\ \partial & x\,\partial & 1 \end{pmatrix}. \tag{6.3}$$

which is row-reduced with row and column orders given by $\delta^{[1]} = (3, 1, 1)$ *and* $\gamma^{[1]} = (3, 2, 1)$ *respectively. Here we have* $\left|\delta^{[1]}\right| < |\delta|$ *but* $\left|\gamma^{[1]}\right| > |\gamma|$*. In addition,* $L^{[1]}$ *is not column-reduced as the resulting leading column coefficient matrix is given by*

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Again, using the construction from Lemma 6.2.2, we multiply $L^{[1]}$ *on the right by*

$$V_1 = \begin{pmatrix} 2 & 0 & 0 \\ -\partial & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

to obtain the column-reduced form

$$L^{[2]} = L^{[1]}\,V_1 = \begin{pmatrix} 2\,x & 2\,\partial^2 & x^2 + x \\ \partial^2 & -\partial & -\partial + 2\,x^2 + 1 \\ -x\,\partial^2 + 2\,\partial & x\,\partial & 1 \end{pmatrix} \tag{6.4}$$

with $\left|\delta^{[2]}\right| = |(2, 2, 2)| > \left|\delta^{[1]}\right|$ *and* $\left|\gamma^{[2]}\right| = |(2, 2, 1)| < \left|\gamma^{[1]}\right|$*. Unfortunately, it is no longer row-reduced so we are back to our first case.* ■

While a single call to row-reduction and then to column-reduction will not necessarily result in a simultaneously row and column reduced form, it turns out that by repeating this process a finite number of times we can always end up with a simultaneously row and column reduced operator. We show this in the following proposition.

Proposition 6.3.1. *Let* $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ *of order* ℓ*. It is always possible to construct two unimodular matrix differential operators* $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ *and* $V \in \mathbb{K}[[x]][\partial]^{n\times n}$ *such that* ULV *is a simultaneously row and column reduced operator.*

Proof. Let us show that by iterating successively row-reduction and column-reduction, we end up with a simultaneously row and column reduced operator. For this, we consider the tuple

$$(r_\ell, c_\ell, r_{\ell-1}, c_{\ell-1}, \ldots, r_0, c_0),$$

where, for $i = 0, \ldots, \ell$, r_i and c_i denote the number of rows and columns of L of order i, respectively with $r_i = 0$, respectively $c_i = 0$, if no such rows, respectively columns, exist. At each step of a row-reduction and of a column-reduction, this tuple strictly decreases in the sense of the lexicographic ordering. Indeed, one step of the row-reduction procedure consists in replacing a row of order i either by a zero row or by a nonzero row of order at most $i-1$. Let $(r_\ell^{[1]}, c_\ell^{[1]}, r_{\ell-1}^{[1]}, c_{\ell-1}^{[1]}, \ldots, r_0^{[1]}, c_0^{[1]})$ and $(r_\ell^{[2]}, c_\ell^{[2]}, r_{\ell-1}^{[2]}, c_{\ell-1}^{[2]}, \ldots, r_0^{[2]}, c_0^{[2]})$ denote the tuples associated with the operators before and after this row operation respectively. Then, for $k = i+1, \ldots, \ell$, we have $r_k^{[2]} = r_k^{[1]}$ and $c_k^{[2]} = c_k^{[1]}$ but $r_i^{[2]} < r_i^{[1]}$ and $c_i^{[2]} \leq c_i^{[1]}$. This implies that

$$(r_\ell^{[2]}, c_\ell^{[2]}, r_{\ell-1}^{[2]}, c_{\ell-1}^{[2]}, \ldots, r_0^{[2]}, c_0^{[2]}) <_{lex} (r_\ell^{[1]}, c_\ell^{[1]}, r_{\ell-1}^{[1]}, c_{\ell-1}^{[1]}, \ldots, r_0^{[1]}, c_0^{[1]})$$

where $<_{lex}$ denotes lexicographic ordering. A similar result holds for column-reduction. Thus, after a finite number of iterations of row-reduction and column-reduction, we get two unimodular matrix differential operators U and V such that ULV is a simultaneously row and column reduced operator. □

We illustrate Proposition 6.3.1 with the following example.

Example 6.3.2. *Consider the matrix differential operator L of order $\ell = 3$ given by (6.2). The tuple $(r_3, c_3, r_2, c_2, r_1, c_1, r_0, c_0)$ associated with L is equal to $(1,1,1,1,1,0,0,1)$. The tuples associated with the operators $L^{[1]}$ and $L^{[2]}$, obtained after applying row-reduction to L and then column-reduction to $L^{[1]}$, are given by $(1,1,0,1,2,1,0,0)$ and $(0,0,3,2,0,1,0,0)$ respectively. We do observe that*

$$(0,0,3,2,0,1,0,0) <_{lex} (1,1,0,1,2,1,0,0) <_{lex} (1,1,1,1,1,0,0,1).$$

We continue now the iteration of row-reduction and column-reduction. Applying row-reduction to $L^{[2]}$ yields a the matrix

$$U_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

such that

$$L^{[3]} = U_2\, L^{[2]} = \begin{pmatrix} 2\,x & 2\,\partial^2 & x^2 + x \\ 2\,\partial & 0 & -x\,\partial + 2\,x^3 + x + 1 \\ -x\,\partial^2 + 2\,\partial & x\,\partial & 1 \end{pmatrix} \tag{6.5}$$

is row-reduced. Since U_2 has polynomial entries, then, according to Corollary 6.2.1, $L^{[3]}$ is simultaneously row and column reduced. So, for L given by (6.2), the iteration stops after the second application of row-reduction. Remark that the tuple associated with $L^{[3]}$ given by $(0,0,2,2,1,1,0,0)$ is, as expected, lower than the one associated with $L^{[2]}$. Finally, L and $L^{[3]}$ are related by $ULV = L^{[3]}$ with

$$U = U_2\, U_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & x & -x\,\partial + 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V = V_1 = \begin{pmatrix} 2 & 0 & 0 \\ -\partial & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

■

A question naturally arises here: given a matrix differential operator L of size $m \times n$ and order ℓ, how many iterations of row-reduction and of column-reduction are needed to get a simultaneously row and column reduced form equivalent to L? When the order ℓ of L is greater than 2, the problem is still open. However, when $\ell = 1$, we will show in the next subsection that we need at most row-column-row reduction steps to get a simultaneously row and column reduced operator equivalent to L.

6.3.2 First-order matrix differential operators

When L is a matrix differential operator of first-order, we have seen in Chapter 5 that computing a simultaneously row and column reduced operator equivalent to L allows us to reduce the DAE $L(y(x)) = f(x)$ into an algebraic system and a first-order system of ODEs. In this subsection, we show that by iterating successively row-reduction and column-reduction on a matrix differential operator of first-order, we are assured of getting a simultaneously row and column reduced operator after at most the second application of row-reduction.

Let L be a matrix differential operator of first-order and rank s. Apply row-reduction to L followed by column-reduction to the resulting row-reduced operator. Operator L is then equivalent to a block diagonal matrix differential operator $\operatorname{diag}\big(L^{[1]}, 0\big)$ with $L^{[1]}$ column-reduced of size $s \times s$ and rank s of the form

$$L^{[1]} = \begin{pmatrix} A_{11}\,\partial + B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

where A_{11} is of size $r_1^{[1]} \times c_1^{[1]}$ and of full column rank (so $c_1^{[1]} \leq r_1^{[1]} \leq s$) and B_{22} of size $r_0^{[1]} \times c_0^{[1]}$ ($r_0^{[1]} = s - r_1^{[1]}$ and $c_0^{[1]} = s - c_1^{[1]}$). Moreover, since $L^{[1]}$ is a square matrix, its leading column coefficient matrix given by

$$\begin{pmatrix} A_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}$$

is invertible which implies that B_{22} is of full row rank.

If $c_1^{[1]} = s$, then $r_1^{[1]} = s$ and $L^{[1]}$ is of the form $L^{[1]} = A_{11}\,\partial + B_{11}$ with invertible matrix A_{11}. Hence, $L^{[1]}$ is simultaneously row and column reduced.

If $c_1^{[1]} = 0$, then $r_1^{[1]} = 0$ (recall that $r_1^{[1]}$ is the number of rows of $L^{[1]}$ of order 1). Thus, $L^{[1]}$ is reduced to B_{22} which is, in this case, invertible (since it is square and of full row rank). Hence, $L^{[1]}$ is simultaneously row and column reduced and we are done.

Otherwise, if $0 \neq c_1^{[1]} = r_1^{[1]} \neq s$ then A_{11} and B_{22} are both invertible (since they are square, of full column rank and full row rank respectively) and hence $L^{[1]}$ is simultaneously row and column reduced.

Finally, if $0 \neq c_1^{[1]} < r_1^{[1]}$, then A_{11} is rectangular of full column rank. It follows that the rank of the leading row coefficient matrix of $L^{[1]}$

$$\begin{pmatrix} A_{11} & 0 \\ B_{21} & B_{22} \end{pmatrix}$$

is necessarily less than s. This means that $L^{[1]}$ is not row-reduced. So let us apply row-reduction to $L^{[1]}$. In this instance, we will find that a unimodular multiplier U, such that $UL^{[1]}$ is row-reduced, is necessarily of order 0, that is, $U \in \mathbb{K}[[x]]^{s \times s}$. Indeed let $(v_1, v_2) \in \mathbb{K}[[x]]^{s \times 1}$ with v_1 and v_2 of sizes $r_1^{[1]} \times 1$ and $r_0^{[1]} \times 1$ respectively, such that

$$(v_1, v_2) \begin{pmatrix} A_{11} & 0 \\ B_{21} & B_{22} \end{pmatrix} = 0.$$

We have then $v_2 B_{22} = 0$ which, since B_{22} is of full row rank, implies $v_2 = 0$ and hence $v_1 \in \ker(A_{11})$. Thus, we construct an invertible matrix $U_1 \in \mathbb{K}[[x]]^{s\times s}$ such that the operator $L^{[2]} = U_1 L^{[1]}$ is of the form

$$L^{[2]} = \begin{pmatrix} \widetilde{A}_{11}\,\partial + \widetilde{B}_{11} & \widetilde{B}_{12} \\ \widetilde{B}_{21} & \widetilde{B}_{22} \end{pmatrix}$$

with a $c_1^{[1]} \times c_1^{[1]}$ invertible matrix $\widetilde{A}_{11}$. As $L^{[1]}$ is column-reduced, then following Corollary 6.2.1, $L^{[2]}$ is so. Now, the fact that $L^{[2]}$ is column-reduced with invertible matrix $\widetilde{A}_{11}$ implies that $\widetilde{B}_{22}$ is also invertible and hence $L^{[2]}$ is simultaneously row and column reduced. Thus, we get a simultaneously row and column reduced form after at most row-column-row reduction steps.

6.4 A second algorithm for simultaneous row and column reduction

In this section, we describe a second algorithm to convert a matrix of differential operator L into one which is simultaneously row and column reduced. To achieve this, let us first give a characterization of a simultaneously row and column reduced operator.

Proposition 6.4.1. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ be a simultaneously row and column reduced matrix differential operator of order ℓ. One can permute the rows and columns of L so that it has the block partition*

$$\left(\begin{array}{ccc|c} L_{11} & \cdots & L_{1k} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ L_{k1} & \cdots & L_{kk} & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right) \tag{6.6}$$

where the L_{ii}'s are square matrices satisfying:

(a) $\ell c(L_{ii})$ is invertible[2]*,*

(b) $\operatorname{ord}(L_{ii}) > \operatorname{ord}(L_{i+1\,i+1})$ for all $i = 1, \ldots, k-1$,

(c) $\operatorname{ord}(L_{ij}) \leq \operatorname{ord}(L_{ii})$ for all $j < i$ and $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{ii})$ for all $j > i$,

(d) $\operatorname{ord}(L_{ij}) \leq \operatorname{ord}(L_{jj})$ for all $i < j$ and $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{jj})$ for all $i > j$.

Conversely, a matrix differential operator of the form (6.6) satisfying conditions (a) to (d) above is simultaneously row and column reduced.

Proof. Let L be a simultaneously row and column reduced matrix differential operator of order ℓ. We first sort the rows and columns of L in a decreasing order. Hence, the zero rows are at the bottom of the matrix and the zero columns are at the end. The nonzero rows and columns of L then form a square submatrix of full rank (this follows from the fact that L is simultaneously row and column reduced and that the row rank and column rank are equal). For ease of presentation, let us assume that L is a square matrix only consisting of these nonzero rows and columns and having decreasing row and column orders.

For $i = 0, \ldots, \ell$, let r_i and c_i denote respectively the number of rows and columns of L of order i with $r_i = 0$, respectively $c_i = 0$, if no such rows, respectively columns, exist. For $i = -1, \ldots, \ell - 1$, let $n_i = r_{i+1} + \cdots + r_\ell$ and $n_\ell = 0$. Note that both r_ℓ and c_ℓ are nonzero since L is of order ℓ. Let L_{11} be the $r_\ell \times c_\ell$ submatrix of L composed of the first r_ℓ rows and c_ℓ

[2] This is equivalent to saying that L_{ii} is row and column reduced with same row and column orders.

columns of L. Then, the leading row and column coefficient matrices of L are respectively given by

$$\begin{pmatrix} \ell c(L_{11}) & 0 \\ * & * \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \ell c(L_{11}) & * \\ 0 & * \end{pmatrix}.$$

Since L is simultaneously row and column reduced, these matrices are respectively of full row rank and full column rank and so $\ell c(L_{11})$ is square (thus, $r_\ell = c_\ell$) and nonsingular.

We assume now that $r_j = c_j$ for $j = \ell, \ldots, i+1$ for a certain $i \leq \ell - 1$ and that L_i, the submatrix of L composed of the first n_i rows and columns of L, can be decomposed into blocks

$$L_i = \begin{pmatrix} L_{11} & \cdots & L_{1u} \\ \vdots & \ddots & \vdots \\ L_{u1} & \cdots & L_{uu} \end{pmatrix}, \quad (u \leq \ell - i)$$

all satisfying conditions (a) to (d) of the proposition ($\operatorname{ord}(L_{uu}) \geq i+1$). Let us show now that $r_i = c_i$. To achieve this, we will first show the following equivalence: $r_i \neq 0 \iff c_i \neq 0$. Assume that $r_i \neq 0$ and $c_i = 0$. It follows that the leading row coefficient matrix of L has the form

$$\begin{pmatrix} \ell c_{\text{row}}(L_i) & 0 \\ * & 0 \\ * & * \end{pmatrix},$$

where $\ell c_{\text{row}}(L_i)$ denotes the leading row coefficient matrix of L_i. This gives a full row rank matrix of size $n_{i-1} \times n_i$ (since L is row-reduced) with $n_i < n_{i-1} = n_i + r_i$, a contradiction. Now, using the fact that L is column-reduced and working similarly with the columns, we show that $c_i \neq 0 \implies r_i \neq 0$. Thus, the equivalence above holds. Assume now that $r_i \neq 0$. If $c_i < r_i$, then, from the leading row coefficient matrix of L, we can extract a full row rank matrix of size $n_{i-1} \times (n_i + c_i)$ with $n_i + c_i < n_i + r_i = n_{i-1}$, a contradiction. Hence, we have $r_i \leq c_i$. Repeating the argument using the fact L is column-reduced gives $c_i \leq r_i$ and so $r_i = c_i$. Let now L_{i-1} denote the $n_{i-1} \times n_{i-1}$ submatrix of L composed of the first n_{i-1} rows and columns of L. If $r_i = 0$ then $L_{i-1} = L_i$. Otherwise, decompose L_{i-1} into blocks as follows

$$L_{i-1} = \left(\begin{array}{ccc|c} & & & L_{1\,u+1} \\ & L_i & & \vdots \\ & & & L_{u\,u+1} \\ \hline L_{u+1\,1} & \cdots & L_{u+1\,u} & L_{u+1\,u+1} \end{array}\right) = \begin{pmatrix} L_{11} & \cdots & L_{1u} & L_{1\,u+1} \\ \vdots & \ddots & \vdots & \vdots \\ L_{u1} & \cdots & L_{uu} & L_{u\,u+1} \\ L_{u+1\,1} & \cdots & L_{u+1\,u} & L_{u+1\,u+1} \end{pmatrix},$$

where $L_{u+1,*} \in \mathbb{K}[[x]][\partial]^{r_i \times n_{i-1}}$, $L_{*,u+1} \in \mathbb{K}[[x]][\partial]^{n_{i-1} \times r_i}$ and $\operatorname{ord}(L_{u+1,*}) = \operatorname{ord}(L_{*,u+1}) = i$. As L_i and L_{i-1} are both row-reduced, it follows that $\operatorname{ord}(L_{u+1\,u+1}) = i$ and $\ell c(L_{u+1\,u+1})$ is invertible (since it is square and of full row rank). Hence, the blocks of L_{i-1} satisfy conditions (a) to (d) of the proposition. The process continues until rows and columns of lowest order are reached.

We will now prove the converse. We consider a matrix differential operator of the form (6.6) satisfying conditions (a) to (d). Its leading row coefficient matrix is then a lower block triangular matrix whose diagonal blocks are $\ell c(L_{11}), \ldots, \ell c(L_{kk})$ and 0. From (b), we can deduce that the leading row coefficient matrix is of full row rank. Using the same arguments on columns, we can deduce that the leading column coefficient matrix is of full column rank. Hence, the operator is simultaneously row and column reduced. □

Example 6.4.1. *Swapping the second and third rows of $L^{[3]}$ given by (6.5), we can partition the resulting matrix differential operator into blocks*

$$\left(\begin{array}{cc|c} 2x & 2\,\partial^2 & x^2 + x \\ -x\,\partial^2 + 2\,\partial & x\,\partial & 1 \\ \hline 2\,\partial & 0 & -x\,\partial + 2x^3 + x + 1 \end{array}\right) = \begin{pmatrix} \widetilde{L}_{11} & \widetilde{L}_{12} \\ \widetilde{L}_{21} & \widetilde{L}_{22} \end{pmatrix},$$

all satisfying conditions (a) to (d) of Proposition 6.4.1. ■

In the sequel, we will take advantage of the previous proposition to describe our second algorithm for converting a matrix differential operator L into a simultaneously row and column reduced one. More precisely, this algorithm brings the operator L into another one having the form (6.6) and satisfying conditions (a) to (d) of Proposition 6.4.1.

By performing column operations, we can ensure that all zero columns of L are to the right and that the nonzero columns are linearly independent. Furthermore, by applying Lemma 6.2.2, we can ensure that the submatrix composed of these nonzero columns is in row-reduced form. The remaining submatrix composed of the nonzero rows and columns is then *nonsingular* (*i.e.*, square and of full rank). From now on, we will work with nonsingular matrix differential operators.

Proposition 6.4.2. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times m}$ be a row-reduced nonsingular matrix differential operator of order ℓ with rows sorted by decreasing order. We can construct an invertible matrix $V \in \mathbb{K}[[x]]^{m\times m}$ such that*

$$L\,V = \begin{pmatrix} L_{11} & \cdots & L_{1k} \\ \vdots & \ddots & \vdots \\ L_{k1} & \cdots & L_{kk} \end{pmatrix} \tag{6.7}$$

where the L_{ii}'s are square matrices satisfying:

(a) $\ell c(L_{ii})$ is invertible,

(b) $\operatorname{ord}(L_{ii}) > \operatorname{ord}(L_{i+1,i+1})$ for all $i = 1, \ldots, k-1$,

(c) $\operatorname{ord}(L_{ij}) \leq \operatorname{ord}(L_{ii})$ for all $j < i$ and $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{ii})$ for all $j > i$.

Proof. For $i = 0, \ldots, \ell$, let r_i denote the number of rows of order i, with $r_i = 0$ if no such rows exist. For $i = -1, \ldots, \ell-1$, let $n_i = r_{i+1} + \cdots + r_\ell$ and $n_\ell = 0$. We will proceed recursively starting with $i = \ell$. The strip of L composed of the first r_ℓ rows is row-reduced. Doing elementary column operations on L (more precisely, multiplying L on the right by an invertible matrix $V_\ell \in \mathbb{K}[[x]]^{m\times m}$), we can ensure that the square submatrix in rows and columns 1 to r_ℓ has a nonsingular leading coefficient matrix and the last $m - r_\ell$ columns of this strip have order at most $\ell - 1$. Suppose now that we have already computed an invertible matrix $V_{i+1} \in \mathbb{K}[[x]]^{m\times m}$ such that the submatrix composed of the first n_i rows of $L\,V_{i+1}$ can be decomposed into blocks satisfying conditions (a) to (c) above. Furthermore, suppose that $r_i \neq 0$. We will now show that there exists an invertible matrix $V_i \in \mathbb{K}[[x]]^{m\times m}$ such that the submatrix formed by the first n_{i-1} rows of $L\,V_i$ can be decomposed into blocks satisfying conditions (a) to (c) of the proposition. Let A_{i+1} denote the square submatrix of $L\,V_{i+1}$ composed of the first n_i rows and columns. Consider the strip of $L\,V_{i+1}$ composed of rows $n_i + 1$ to n_{i-1}. Since $L\,V_{i+1}$ and A_{i+1} are both row-reduced[3], it follows that the submatrix formed by columns from $n_i + 1$ to m of this strip is of order i and row-reduced, that is, its leading row coefficient matrix is of full row rank. Carrying out elementary column operations on $L\,V_{i+1}$, we can ensure that the square submatrix B_i in rows and columns $n_i + 1$ to n_{i-1} is of order i and has a nonsingular leading coefficient matrix, and that the last $m - n_{i-1}$ columns of this strip have order at most $i - 1$. Note that the column operations performed above only require elements from $\mathbb{K}[[x]]$ as we are always eliminating only by means of entries from the leading row coefficient matrix. Thus, there

[3] $L\,V_{i+1}$ is row-reduced due to Corollary 6.2.1.

exists an invertible matrix $W_i \in \mathbb{K}[[x]]^{m\times m}$ such the submatrix of $L\,V_{i+1}\,W_i$ composed of the first n_{i-1} rows is of the form

$$\begin{pmatrix} A_{i+1} & * & * \\ * & B_i & * \end{pmatrix}$$

and has a leading row coefficient matrix

$$\begin{pmatrix} \ell c_{\text{row}}(A_{i+1}) & 0 & 0 \\ * & \ell c(B_i) & 0 \end{pmatrix}.$$

Let $V_i = V_{i+1}\,W_i$. Thus, the first n_{i-1} rows of $L\,V_i$ can be decomposed into blocks satisfying conditions (a) to (c) above. We continue then the process to reach the rows of lowest order. □

Remark 6.4.1. *For a given strip of same row order, the procedure requires that we ensure that the $r_i \times r_i$ matrix starting at column $n_i + 1$ has nonsingular leading coefficient and that the last $m - n_{i-1}$ columns of this strip have order at most $i - 1$. This can be accomplished by converting the leading row coefficient matrix of this strip from columns $n_i + 1$ to m (which is of full row rank) into column echelon form.*

Example 6.4.2. *Applying the procedure described in the proof of Proposition 6.4.2 to a matrix differential operator having blocks of orders 6, 4, 2 and 1 as in*

$$\begin{pmatrix} 6 & 6 & 6 & 6 \\ 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

the resulting orders would then become

$$\begin{pmatrix} 6 & 5 & 5 & 5 \\ 4 & 4 & 3 & 3 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

■

Remark 6.4.2. *Note that we can continue the process described in the proof of Proposition 6.4.2 (again working top to bottom) to eliminate the leading coefficient matrices of the blocks to the left of any diagonal block so that these blocks have smaller order, that is, condition (c) is replaced by*

(c') $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{ii})$ *for all* $j \neq i$.

Example 6.4.3. *Continuing with the previous example, we can use the leading coefficient matrix in each diagonal block to reduce the orders of the blocks to the left. Thus, by performing elementary column operations, the orders can be reduced to*

$$\begin{pmatrix} 6 & 5 & 5 & 5 \\ 3 & 4 & 3 & 3 \\ 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

if so desired. ■

The construction in Proposition 6.4.2 does not necessarily result in a simultaneously row and column reduced form. Fortunately, Proposition 6.4.1 provides the required orders for a simultaneously row and column reduced matrix differential operator.

Definition 6.4.1. *Let $L = (L_{ij})_{1\le i,j\le k} \in \mathbb{K}[[x]][\partial]^{m\times m}$ be a matrix differential operator satisfying conditions (a) to (c) of Proposition 6.4.2. The* defect *of the block row $L_{i,*}$ for $i = 1,\dots,k-1$ is defined by*

$$\operatorname{defect}(L)_i = \max\{\operatorname{ord}(L_{ij}) - \operatorname{ord}(L_{jj}),\ j = i+1,\dots,k\}\,.$$

Thus, a nonsingular row-reduced matrix differential operator $L = (L_{ij})_{1\le i,j\le k}$ satisfying conditions (a) to (c) of Proposition 6.4.2 is simultaneously row and column reduced if and only if $\operatorname{defect}(L)_i \le 0$ for $i = 1,\dots,k-1$.

Proposition 6.4.3. *Let $L = (L_{ij})_{1\le i,j\le k} \in \mathbb{K}[[x]][\partial]^{m\times m}$ be a row-reduced nonsingular matrix differential operator satisfying conditions (a) to (c) of Proposition 6.4.2. Then, we can construct a unimodular matrix $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ such that UL has the block partition $\left(\overline{L}_{ij}\right)_{1\le i,j\le k}$, where the blocks $\overline{L}_{ij}$ are of the same size as the L_{ij}'s and satisfy conditions (a) to (d) of Proposition 6.4.1.*

Proof. Since L already satisfies conditions (a) to (c) of Proposition 6.4.2, it remains to perform row operations so that the condition (d) of Proposition 6.4.1 can be also satisfied. This is achieved by reducing the positive defects of the block rows of L to 0 proceeding from the bottom to the top. Suppose that $\operatorname{defect}(L)_i \le 0$ for $i = i_0+1,\dots,k-1$ and $\operatorname{defect}(L)_{i_0} > 0$. We will explain how one can lower the defect of the i_0th block row of L. Let j_0 be the smallest integer j for which $\operatorname{ord}(L_{i_0 j}) - \operatorname{ord}(L_{jj}) = \operatorname{defect}(L)_{i_0}$. We first lower the order of the block $L_{i_0 j_0}$ in the following way. Compute the adjoint[4] $\operatorname{adj}(\ell c(L_{j_0 j_0}))$ of $\ell c(L_{j_0 j_0})$. Then replace the i_0th block row $L_{i_0,*}$ of L by

$$\det(\ell c(L_{j_0 j_0}))\, L_{i_0,*} - \ell c(L_{i_0 j_0})\operatorname{adj}(\ell c(L_{j_0 j_0}))\,\partial^{\alpha} L_{j_0,*}, \tag{6.8}$$

where $\alpha = \operatorname{defect}(L)_{i_0}$. This is performed by multiplying L on the left by a unimodular matrix differential operator in $\mathbb{K}[[x]][\partial]^{m\times m}$. Let $\widetilde{L} = \left(\widetilde{L}_{ij}\right)_{1\le i,j\le k}$ denote the resulting matrix differential operator. Then one can check that

$$\operatorname{ord}(\widetilde{L}_{i_0 j}) - \operatorname{ord}(\widetilde{L}_{jj}) \begin{cases} < \operatorname{defect}(L)_{i_0} & \text{for} \quad j \le j_0 \\ \le \operatorname{defect}(L)_{i_0} & \text{for} \quad j > j_0 \end{cases},$$

and so, $\operatorname{defect}(\widetilde{L})_{i_0} \le \operatorname{defect}(L)_{i_0}$. Two cases then arise:

1. $\operatorname{defect}(\widetilde{L})_{i_0} < \operatorname{defect}(L)_{i_0}$, in which case we are done, or
2. $\operatorname{defect}(\widetilde{L})_{i_0} = \operatorname{defect}(L)_{i_0}$. In this case, the smallest integer j for which $\operatorname{ord}(\widetilde{L}_{i_0 j}) - \operatorname{ord}(\widetilde{L}_{jj}) = \operatorname{defect}(\widetilde{L})_{i_0}$ is greater than j_0. Hence, the "value of j_0" increases, and so after a finite number of iterations, the defect of the i_0th block row will decrease.

Iterating this process, we reduce all positive defects to zero and hence we get an operator with block partition satisfying conditions (a) to (d) of Proposition 6.4.1. □

Remark 6.4.3. *The elimination step in equation (6.8) can be viewed as the block row operation which replaces the i_0th block row $L_{i_0,*}$ of L by*

$$L_{i_0,*} - \ell c(L_{i_0 j_0})\,(\ell c(L_{j_0 j_0}))^{-1}\,\partial^{\alpha} L_{j_0,*}.$$

However, for computational purposes it is better to work in the ring $\mathbb{K}[[x]]$ rather than in its quotient field $\mathbb{K}((x))$. Elimination in this case implies solving the linear system of equations

$$X_{i_0}\,\ell c(L_{j_0 j_0}) = \ell c(L_{i_0 j_0})$$

[4] The transpose of the cofactor matrix.

for X_{i_0} a matrix of the same size as $L_{i_0 j_0}$. We can solve such a system and remain in the domain $\mathbb{K}[[x]]$ by using fraction-free Gaussian elimination (see for example [48, Chapter 9]). However this produces a solution for the system

$$\det(\ell c(L_{j_0 j_0}))\, X_{i_0} = \ell c(L_{i_0 j_0})\ \mathrm{adj}(\ell c(L_{j_0 j_0}))$$

and hence replacing the i_0th block row $L_{i_0,*}$ of L is done via equation (6.8). We remark that in order to minimize growth of coefficients for a given row - say row $\hat{i}$ of block i_0 - one still needs to remove common factors of the elimination terms, which in this case means removing the greatest common factor of the terms in row $\hat{i}$.

Example 6.4.4. *Consider the matrix differential operator having order bounds*

$$\begin{pmatrix} 6 & 5 & 5 & 5 \\ 4 & 4 & 3 & 3 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

as in Example 6.4.2. Suppose that we have first reduced the defects of the third and then the second block rows to 0 and get order bounds

$$\begin{pmatrix} 6 & 5 & 5 & 5 \\ 4 & 4 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Let us now explain how to reduce the defect of the first block row $i_0 = 1$. Assuming all orders are attained, then the defect of the first block row is 4 and is attained for $j_0 = 4$. We reduce the order of block $(1,4)$ using the fourth block row. We obtain a matrix of orders of the form

$$\begin{pmatrix} 6 & 5 & 5 & 4 \\ 4 & 4 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

The defect of the first block row is now 3 and it is attained for $j = 3, 4$. We repeat the reduction process, first by reducing the order of block entry $(1,3)$ using the third block row and then reducing the order of block entry $(1,4)$ by the fourth block row. This gives a matrix having order bounds

$$\begin{pmatrix} 6 & 5 & 4 & 3 \\ 4 & 4 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

The first block row is now of defect 2. Thus, by continuing the process above, we reduce the defect to 0 giving order bounds

$$\begin{pmatrix} 6 & 4 & 2 & 1 \\ 4 & 4 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

■

We summarize our result in the following theorem:

Theorem 6.4.1. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times m}$ be a row-reduced nonsingular matrix differential operator. One can construct a unimodular matrix differential operator $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ and an invertible matrix $V \in \mathbb{K}[[x]]^{m\times m}$ such that ULV has the block partition $(L_{ij})_{1\leq i,j\leq k}$, where the blocks L_{ij} satisfy conditions (a) to (d) of Proposition 6.4.1.*

More generally,

Theorem 6.4.2. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ be a matrix differential operator. One can construct two unimodular matrices $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ and $V \in \mathbb{K}[[x]][\partial]^{n\times n}$ such that ULV has the form (6.6) with conditions (a) to (d) of Proposition 6.4.1 satisfied.*

We illustrate the second algorithm with the following example.

Example 6.4.5. *We will apply here the second algorithm to the matrix differential operator L given by (6.2). Notice that L is already column-reduced and hence it is of rank 3 (so L is nonsingular). Applying row-reduction to L, we have got the row-reduced operator $L^{[1]}$ given by (6.3) which is nonsingular as well. Furthermore, the latter operator partitioned into blocks as follows*

$$L^{[1]} = \left(\begin{array}{c|cc} \partial^3+x & 2\,\partial^2 & x^2+x \\ \hline 0 & -\partial & -\partial+2x^2+1 \\ \partial & x\,\partial & 1 \end{array}\right)$$

satisfies already conditions (a) to (c) in Proposition 6.4.2. To obtain a simultaneously row and column reduced operator, we need to reduce the order of the block in position $(1,2)$. We can achieve this using the diagonal block $(2,2)$. Indeed, the defect of the first block row of $L^{[1]}$ is equal to 1. Proceeding as in the proof of Proposition 6.4.3 and using elementary row operations, we can replace $L^{[1]}_{1,*}$ by

$$\begin{aligned} x\,L^{[1]}_{1,*} - \begin{pmatrix}2 & 0\end{pmatrix}\begin{pmatrix}0 & 1\\ -x & -1\end{pmatrix}\partial L^{[1]}_{2,*} &= \begin{pmatrix}x\,\partial^3+x^2 & 2\,x\,\partial^2 & x^3+x^2\end{pmatrix} - \begin{pmatrix}2\,\partial^2 & 2\,x\,\partial^2+2\,\partial & 2\,\partial\end{pmatrix} \\ &= \begin{pmatrix}x\,\partial^3-2\,\partial^2+x^2 & -2\,\partial & -2\,\partial+x^3+x^2\end{pmatrix}. \end{aligned}$$

Hence, L is equivalent to the operator

$$\left(\begin{array}{c|cc} x\,\partial^3-2\,\partial^2+x^2 & -2\,\partial & -2\,\partial+x^3+x^2 \\ \hline 0 & -\partial & -\partial+2x^2+1 \\ \partial & x\,\partial & 1 \end{array}\right) = \begin{pmatrix}\widehat{L}_{11} & \widehat{L}_{12}\\ \widehat{L}_{21} & \widehat{L}_{22}\end{pmatrix}$$

which is simultaneously row and column reduced. Remark that the latter operator is different from the simultaneously row and column reduced operator (equivalent to L) given in Example 6.4.1 obtained by applying iteratively the procedures row-reduction and column-reduction. They are even of different orders. Nevertheless, we point out that the sum of the orders of diagonal blocks $\widehat{L}_{ii}$ multiplied respectively by their size, i.e., $\operatorname{ord}(\widehat{L}_{11}) + 2\operatorname{ord}(\widehat{L}_{22})$, *is equal to the one corresponding to the diagonal blocks found in Example 6.4.1,* i.e., $2\operatorname{ord}(\widetilde{L}_{11}) + \operatorname{ord}(\widetilde{L}_{22})$. ■

We will now define the *two-sided block Popov form*. We have called it so since it is similar to the Popov form of Ore matrix polynomials (see for instance [40]).

Definition 6.4.2. *Let $L \in \mathbb{K}[[x]][\partial]^{m\times m}$ be a nonsingular matrix differential operator. We say that L is* in two-sided block Popov form *if L has the block partition*

$$L = \begin{pmatrix} L_{11} & \cdots & L_{1k} \\ \vdots & \ddots & \vdots \\ L_{k1} & \cdots & L_{kk} \end{pmatrix},$$

where the L_{ii}'s are square matrices satisfying:

(a) $\ell c(L_{ii})$ *is invertible,*

(b) $\operatorname{ord}(L_{ii}) > \operatorname{ord}(L_{i+1,i+1})$ *for all* $i = 1, \ldots, k-1$,

(c) $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{ii})$ *for all* $j = 1, \ldots, k$ *such that* $j \neq i$,

(d) $\operatorname{ord}(L_{ij}) < \operatorname{ord}(L_{jj})$ *for all* $i = 1, \ldots, k$ *such that* $i \neq j$.

Corollary 6.4.1. *Let* $L \in \mathbb{K}[[x]][\partial]^{m\times m}$ *be a row-reduced nonsingular matrix differential operator. Then there exist a unimodular matrix differential operator* $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ *and an invertible matrix* $V \in \mathbb{K}[[x]]^{m\times m}$ *such that* ULV *is in two-sided block Popov form.*

Proof. From Remark 6.4.2, we can ensure that all blocks to the left of a diagonal block have lower orders using only column operations. The procedure in the proof of Proposition 6.4.3 can be extended to ensure that all the blocks above a diagonal one have lower orders as well. □

Example 6.4.6. *If our starting matrix was the one in Example 6.4.3 to which we applied the same row operations as in Example 6.4.4, then we would have a matrix of orders bounded by*

$$\begin{pmatrix} 6 & 4 & 2 & 1 \\ 3 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

If our procedure produced defects of -1 *rather than* 0 *at every block row, then we would get*

$$\begin{pmatrix} 6 & 3 & 1 & 0 \\ 3 & 4 & 1 & 0 \\ 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and hence the resulting matrix differential operator is in two-sided block Popov form. ■

Corollary 6.4.2. *Let* $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ *be a matrix differential operator. Then there exist two unimodular matrix differential operator* $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ *and* $V \in \mathbb{K}[[x]][\partial]^{n\times n}$ *such that* $ULV = \operatorname{diag}(\widetilde{L}, 0)$, *where* $\widetilde{L}$ *is in two-sided block Popov form.*

The difference between the two-sided block Popov form and the Popov form is that the former is a square matrix operator obtained by performing elementary operations on both rows and columns of the input operator, while the latter is, in general, rectangular computed by working on either rows or columns of the input operator but not on both of them at the same time.

6.5 Algorithms and complexity

In this section, we give the algorithmic descriptions of the methods described in Propositions 6.4.2 and 6.4.3 with their complexities.

ALGORITHM **Block_Reduced_Form**

INPUT: $L \in \mathbb{K}[[x]][\partial]^{m\times m}$ a nonsingular row-reduced matrix differential operator of order ℓ.
OUTPUT: An invertible matrix $V \in \mathbb{K}[[x]]^{m\times m}$ (over $\mathbb{K}((x))$) and a matrix differential operator $\widetilde{L}$ s.t. $\widetilde{L} = L\,V$ and $\widetilde{L}$ can be partitioned into blocks $\widetilde{L}_{ij}$ for $1 \le i, j \le k$ satisfying conditions (a) to (c) of Proposition 6.4.2.

INITIALIZATION: Let $V = I_m$ and $\widetilde{L} = L$.
Let r_i, for $i = 0, \ldots, \ell$, denote the number of rows of $\widetilde{L}$ of order i.
Define $n_\ell = 0$ and $n_i = \sum_{j=i+1}^{\ell} r_j$ for $i = -1, \ldots, \ell - 1$.

1. Sort rows of $\widetilde{L}$ in a decreasing order;
2. **For** i from ℓ by -1 to 0 **do**
   2.1. Define $\widetilde{L}_0$ as the leading row coefficient matrix of $\widetilde{L}$;
   2.2. Let B_i denote the submatrix of $\widetilde{L}_0$ composed of rows $n_i + 1$ to n_{i-1} and columns $n_i + 1$ to m;
   2.3. Compute an invertible matrix $V_i \in \mathbb{K}[[x]]^{(m-n_i)\times(m-n_i)}$ such that $B_i\,V_i$ is in column echelon form;
   2.4. Let $\widetilde{L} = \widetilde{L}\,\mathrm{diag}(I_{n_i}, V_i)$ and $V = V\,\mathrm{diag}(I_{n_i}, V_i)$;
   end do;
3. **Return** V and $\widetilde{L}$;

Proposition 6.5.1. *Algorithm* **Block_Reduced_Form** *can be done using* $O\big(m^3\,\ell^3\big)$ *operations in* $\mathbb{K}[[x]]$.

Proof. We first consider the cost of one passage of the **For** loop, say at index $i \in \{0, \ldots, \ell\}$. In this case, Step 2.3 requires at most $O\big(r_i\,(m - n_i)^2\big)$ operations in $\mathbb{K}[[x]]$. In Step 2.4, the product $V\,\mathrm{diag}(I_{n_i}, V_i)$ can be done in at most $O\big(m\,(m - n_i)^2\big)$ operations in $\mathbb{K}[[x]]$. In order to determine the cost of the product $\widetilde{L}\,\mathrm{diag}(I_{n_i}, V_i)$ we can write the operator $\widetilde{L}$ as

$$\widetilde{L} = \widetilde{A}_\ell(x)\,\partial^\ell + \widetilde{A}_{\ell-1}(x)\,\partial^{\ell-1} + \cdots + \widetilde{A}_0(x),$$

where $\widetilde{A}_j(x) \in \mathbb{K}[[x]]^{m\times m}$ for $j = 0, \ldots, \ell$, and then observe that

$$\partial^j\,\mathrm{diag}(I_{n_i}, V_i) = \mathrm{diag}\left(I_{n_i}\,\partial^j, \sum_{s=0}^{j} \binom{j}{s} \partial^s(V_i)\,\partial^{j-s}\right).$$

Thus, the cost of one product of the form $\widetilde{A}_j(x)\,\partial^j\,\mathrm{diag}(I_{n_i}, V_i)$ is equal to the cost of $j + 1$ products of an $m \times (m - n_i)$ matrix by an $(m - n_i) \times (m - n_i)$ matrix. Therefore, the product $\widetilde{L}\,\mathrm{diag}(I_{n_i}, V_i)$ can be done in at most $O\Big(\sum_{j=0}^{\ell}(j + 1)\,m\,(m - n_i)^2\Big) = O\big(m\,\ell^2\,(m - n_i)^2\big)$ operations in $\mathbb{K}[[x]]$. Hence, one passage of the **For** loop can be done in at most $O\big(m\,\ell^2\,(m - n_i)^2\big)$ operations in $\mathbb{K}[[x]]$. Since $m - n_i \le m$ and the **For** loop is repeated $\ell + 1$ times, Algorithm **Block_Reduced_Form** returns V and $\widetilde{L}$ using at most $O\big(m^3\,\ell^3\big)$ operations in $\mathbb{K}[[x]]$. □

ALGORITHM **SRCR_Form**

INPUT: $L = (L_{ij})_{1\le i,j\le k} \in \mathbb{K}[[x]][\partial]^{m\times m}$ a nonsingular matrix differential operator of order ℓ where the $L_{ij} \in \mathbb{K}[[x]][\partial]^{m_i\times m_j}$ satisfy conditions (a) to (c) of Proposition 6.4.2.

OUTPUT: A unimodular matrix differential operator $U \in \mathbb{K}[[x]][\partial]^{m\times m}$ and a matrix differential operator $\widetilde{L} = (\widetilde{L}_{ij})_{1\le i,j\le k} \in \mathbb{K}[[x]][\partial]^{m\times m}$ where the blocks $\widetilde{L}_{ij} \in \mathbb{K}[[x]][\partial]^{m_i\times m_j}$ satisfy conditions (a) to (d) of Proposition 6.4.1 and such that $\widetilde{L} = UL$.

INITIALIZATION: Let $\widetilde{L} = L$, respectively $U = I_m$, partitioned into blocks $(\widetilde{L}_{ij})_{1\le i,j\le k}$, respectively $(U_{ij})_{1\le i,j\le k}$ as L.

For i from $k-1$ by -1 to 1 **do**

1. Define $\operatorname{defect}(\widetilde{L})_i = \max\{\operatorname{ord}(\widetilde{L}_{ij}) - \operatorname{ord}(\widetilde{L}_{jj}),\ j = i+1,\dots,k\}$;
2. Let $W = I_m$ partitioned into blocks $(W_{ij})_{1\le i,j\le k}$ as $\widetilde{L}$;
3. **While** $\operatorname{defect}(\widetilde{L})_i > 0$ **do**
   - 3.1. Define $j_0 = \min\Big\{j \in \{i+1,\dots,k\};\ \operatorname{ord}(\widetilde{L}_{ij}) - \operatorname{ord}(\widetilde{L}_{jj}) = \operatorname{defect}(\widetilde{L})_i\Big\}$;
   - 3.2. Define $\alpha = \operatorname{defect}(\widetilde{L})_i$;
     // We avoid fractions in elimination Steps 3.3 and 3.4
   - 3.3. Let $\widetilde{L}_{i,*} = \det(\ell c(\widetilde{L}_{j_0j_0}))\,\widetilde{L}_{i,*} - \ell c(\widetilde{L}_{ij_0})\operatorname{adj}(\ell c(\widetilde{L}_{j_0j_0}))\,\partial^\alpha \widetilde{L}_{j_0,*}$;
   - 3.4. Let $W_{i,*} = \det(\ell c(\widetilde{L}_{j_0j_0}))\,W_{i,*} - \ell c(\widetilde{L}_{ij_0})\operatorname{adj}(\ell c(\widetilde{L}_{j_0j_0}))\,\partial^\alpha W_{j_0,*}$;
   - 3.5. Update $\operatorname{defect}(\widetilde{L})_i$;

   end do;
4. Let $U = W\,U$;

end do;
Return $\widetilde{L}$ and U;

Proposition 6.5.2. *Algorithm* **SRCR_Form** *can be done using* $O\big(k^3\,m^3\,\ell^2\big)$ *operations in* $\mathbb{K}[[x]]$ *with* $k \le \min\{m, \ell+1\}$.

Proof. First, we note that, during the algorithm, the leading coefficient matrices of the diagonal blocks $\widetilde{L}_{jj}$ for $j = 1,\dots,k$ remain unchanged. Therefore, we need only to compute once and for all the determinants and the adjoints (hence the inverses) of the matrices $\ell c(\widetilde{L}_{jj})$ for $j = 2,\dots,k$. The blocks $\widetilde{L}_{jj}$ are of size $m_j \times m_j$ and so computing the determinants and adjoints can be done in at most $O\Big(\sum_{j=2}^k m_j^3\Big) = O\big(m^3\big)$ operations in $\mathbb{K}[[x]]$ since $\sum_{j=2}^k m_j \le m$. Let us now study the cost of the **While** loop, starting with the cost of Step 3.3. Multiplying $\ell c(\widetilde{L}_{ij_0})$ by $\operatorname{adj}(\ell c(\widetilde{L}_{j_0j_0}))$ can be done in at most $O\Big(m_i\,m_{j_0}^2\Big)$ operations in $\mathbb{K}[[x]]$. Consider now the cost of multiplying the operator $\partial^\alpha \widetilde{L}_{j_0,*}$ on the left by the matrix $\ell c(\widetilde{L}_{ij_0})\operatorname{adj}(\ell c(\widetilde{L}_{j_0j_0}))$. Note that, since $\operatorname{ord}(\widetilde{L}_{j_0,*}) = \operatorname{ord}(\widetilde{L}_{j_0j_0})$, we have

$$\operatorname{ord}(\partial^\alpha \widetilde{L}_{j_0,*}) = \alpha + \operatorname{ord}(\widetilde{L}_{j_0,*}) = \operatorname{ord}(\widetilde{L}_{ij_0}) - \operatorname{ord}(\widetilde{L}_{j_0j_0}) + \operatorname{ord}(\widetilde{L}_{j_0,*}) = \operatorname{ord}(\widetilde{L}_{ij_0}).$$

Thus, the order of the operator $\partial^\alpha \widetilde{L}_{j_0,*}$ is at most ℓ. Therefore, the cost of multiplying $\partial^\alpha \widetilde{L}_{j_0,*}$ on the left by $\ell c(\widetilde{L}_{ij_0})\operatorname{adj}(\ell c(\widetilde{L}_{j_0j_0}))$ is equivalent to the cost of at most $\ell+1$ products of an $m_i \times m_{j_0}$ matrix by an $m_{j_0} \times m$ matrix with entries in $\mathbb{K}[[x]]$ which can be done using $O(m_i\,m_{j_0}\,m\,\ell)$ operations in $\mathbb{K}[[x]]$. Since the order of $\widetilde{L}_{i,*}$ is always bounded by ℓ, multiplying the block row $\widetilde{L}_{i,*}$ by $\det(\ell c(\widetilde{L}_{j_0j_0}))$ in Step 3.3 can be done using at most $O(m_i\,m\,\ell)$ operations

in $\mathbb{K}[[x]]$. Thus, Step 3.3 can be done in at most $O\big(m^3\,\ell\big)$ operations in $\mathbb{K}[[x]]$. The cost of Step 3.4 is seen to be the cost of multiplying $W_{i,*}$ by $\det(\ell c(\widetilde{L}_{j_0 j_0}))$ since the block row $W_{j_0,*}$ is always equal to

$$W_{j_0,*} = \begin{pmatrix} 0 & \cdots & 0 & I_{m_{j_0}} & 0 & \cdots & 0 \end{pmatrix}.$$

Let $\alpha_i^0 \leq \ell$ denote the defect of the block row $L_{i,*}$. Since $W_{j_0,*}$ is always of the form above, the order of $W_{i,*}$ is then bounded by α_i^0. Thus, Step 3.4 costs at most $O(m_i\,m\,\ell)$ operations in $\mathbb{K}[[x]]$. One passage of the **While** loop can then be done in at most $O\big(m^3\,\ell\big)$ operations in $\mathbb{K}[[x]]$. To reduce the defect of the ith block row to zero, Steps 3.1 to 3.5 are repeated at most $(k-i)\,\alpha_i^0$ times. Since $(k-i)\,\alpha_i^0 \leq k\,\ell$, Step 3 can be done in at most $O\big(k\,m^3\,\ell^2\big)$ operations in $\mathbb{K}[[x]]$. It remains to determine the cost of Step 4. Due to the particular structure of W, the product $W\,U$ can be reduced to replace the ith block row of U by $\sum_{j=i}^{k} W_{ij}\,U_{j,*}$ while keeping the other block rows of U unchanged. Thus, we need to study the cost of $\sum_{j=i}^{k} W_{ij}\,U_{j,*}$. Remark that the kth block row of U is always equal to

$$U_{k,*} = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & I_{m_k} \end{pmatrix},$$

and so by induction we can show that for $j = i+1, \ldots, k-1$, we have $\operatorname{ord}(U_{j,*}) \leq \sum_{s=j}^{k-1} \alpha_s^0$ where α_s^0 denotes the defect of the block row $L_{s,*}$. Consider now the cost of multiplying operator W_{ij} by $U_{j,*}$. Since $W_{ij} \in \mathbb{K}[[x]][\partial]^{m_i \times m_j}$ is of order bounded by $\alpha_i^0 \leq \ell$ and $U_{j,*} \in \mathbb{K}[[x]][\partial]^{m_j \times m}$ is of order bounded by $\sum_{s=j}^{k-1} \alpha_s^0 \leq k\,\ell$, one product of the form $W_{ij}\,U_{j,*}$ costs at most $O\big(k\,m^3\,\ell^2\big)$ operations in $\mathbb{K}[[x]]$. Step 4 can thus be done in at most $O\big(k^2\,m^3\,\ell^2\big)$ operations in $\mathbb{K}[[x]]$. As Steps 3 and 4 are repeated $k-1$ times, the algorithm returns a simultaneously row and column reduced operator equivalent to the input L using at most $O\big(k^3\,m^3\,\ell^2\big)$ operations in $\mathbb{K}[[x]]$. □

6.6 Reduction of higher-order linear differential systems

In this section, we consider a nonsingular matrix differential operator L which is simultaneously row and column reduced. Consequently, L can be partitioned into blocks L_{ij} for $i, j = 1, \ldots, k$ satisfying conditions (a) to (d) of Proposition 6.4.1. We are interested in the case where L_{kk} has order 0. In particular, we will show that when applied to linear differential systems of the form (6.1) then the problem can be decoupled into separate purely differential and purely algebraic problems.

Proposition 6.6.1. *Let L be a nonsingular matrix differential operator partitioned into blocks L_{ij} with $1 \leq i, j \leq k$ and $k \geq 2$ satisfying conditions (a) to (d) of Proposition 6.4.1. Assume that* $\operatorname{ord}(L_{kk}) = 0$. *Then, there exist two invertible matrices U and V with entries in $\mathbb{K}[[x]]$ such that ULV is of the form*

$$\begin{pmatrix} \widetilde{L} & 0 \\ 0 & L_{kk} \end{pmatrix} = \left(\begin{array}{ccc|c} \widetilde{L}_{11} & \cdots & \widetilde{L}_{1\,k-1} & 0 \\ \vdots & & \vdots & \vdots \\ \widetilde{L}_{k-1\,1} & \cdots & \widetilde{L}_{k-1\,k-1} & 0 \\ \hline 0 & \cdots & 0 & L_{k\,k} \end{array}\right),$$

where the blocks $\widetilde{L}_{ij}$ satisfy conditions (a) to (d) of Proposition 6.4.1 and $\operatorname{ord}(\widetilde{L}_{ii}) = \operatorname{ord}(L_{ii})$ *for $1 \leq i \leq k-1$.*

Proof. Since we assume that $\operatorname{ord}(L_{kk}) = 0$, L_{kk} is then an invertible matrix with entries in $\mathbb{K}[[x]]$. Hence, by means of elementary row and column operations, we can eliminate all the blocks above and to the left of L_{kk} without affecting the orders of the diagonal blocks nor the invertibility of their leading coefficient matrices. □

In the case of a linear differential system given by

$$L(y(x)) = f(x),$$

we can separate such systems into purely differential and purely algebraic systems. Indeed, using Proposition 6.6.1, we see that the system can be transformed as

$$\begin{cases} \widetilde{L}(w_1(x)) = h_1(x) \\ L_{kk}\, w_2(x) = h_2(x) \end{cases},$$

where

$$\begin{pmatrix} w_1(x) \\ w_2(x) \end{pmatrix} = V^{-1} y(x) \quad \text{and} \quad \begin{pmatrix} h_1(x) \\ h_2(x) \end{pmatrix} = U f(x).$$

Additionally, system $\widetilde{L}(w_1(x)) = h_1(x)$ can be converted into a first-order system of ordinary differential equations as it is stated by the following proposition.

Proposition 6.6.2. *Let L be a nonsingular matrix differential operator partitioned into blocks L_{ij} with $1 \leq i, j \leq k$ satisfying conditions (a) to (d) of Proposition 6.4.1. Assume that $\operatorname{ord}(L_{kk}) > 0$. For $i = 1, \ldots, k$, let m_i denote the size of the block L_{ii}. The differential system $L(y(x)) = f(x)$ can then be converted into a first-order system of ordinary differential equations of size $\sum_{i=1}^{k} m_i \operatorname{ord}(L_{ii}) \times \sum_{i=1}^{k} m_i \operatorname{ord}(L_{ii})$.*

Example 6.6.1. *Suppose that L is of the form*

$$L = \begin{pmatrix} A_{11}(x)\,\partial^3 + B_{11}(x)\,\partial^2 + C_{11}(x)\,\partial + D_{11}(x) & B_{12}(x)\,\partial^2 + C_{12}(x)\,\partial + D_{12}(x) \\ B_{21}(x)\,\partial^2 + C_{21}(x)\,\partial + D_{21}(x) & B_{22}(x)\,\partial^2 + C_{22}(x)\,\partial + D_{22}(x) \end{pmatrix}$$

where A_{11} and B_{22} are two invertible matrices of sizes $m_1 \times m_1$ and $m_2 \times m_2$, respectively. Consider a differential system of the form $L(y(x)) = f(x)$ with $y(x)$ and $f(x)$ partitioned into blocks as L, that is,

$$y(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad \text{and} \quad f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}.$$

Then, $L(y(x)) = f(x)$ can be converted into a system of ordinary differential equations of first-order and size $3\,m_1 + 2\,m_2$ of the form $\widetilde{L}(Y(x)) = G(x)$ with

$$\widetilde{L} = \begin{pmatrix} I_{m_1} & 0 & 0 & 0 & 0 \\ 0 & I_{m_1} & 0 & 0 & 0 \\ 0 & 0 & A_{11}(x) & 0 & B_{12}(x) \\ 0 & 0 & 0 & I_{m_2} & 0 \\ 0 & 0 & 0 & 0 & B_{22}(x) \end{pmatrix} \partial + \begin{pmatrix} 0 & -I_{m_1} & 0 & 0 & 0 \\ 0 & 0 & -I_{m_1} & 0 & 0 \\ D_{11}(x) & C_{11}(x) & B_{11}(x) & D_{12}(x) & C_{12}(x) \\ 0 & 0 & 0 & 0 & -I_{m_2} \\ D_{21}(x) & C_{21}(x) & B_{21}(x) & D_{22}(x) & C_{22}(x) \end{pmatrix},$$

$$Y(x) = \begin{pmatrix} y_1(x) \\ y_1'(x) \\ y_1''(x) \\ y_2(x) \\ y_2'(x) \end{pmatrix} \quad \text{and} \quad G(x) = \begin{pmatrix} 0 \\ 0 \\ f_1(x) \\ 0 \\ f_2(x) \end{pmatrix},$$

where y_1'' denotes the second derivative of y_1. ■

Conclusion and Perspectives of Part II

In this part of the thesis, we have developed new two-sided row and column reduction algorithms for matrix differential operators with formal power series coefficients. When applied to linear differential-algebraic equations of arbitrary order, such a reduction decouples the input system into separately differential and algebraic components. Our algorithm leads to a complete mastering of the first-order case: reducing a linear DAE of first-order to a system of ordinary differential equations of first-order and a system of algebraic equations. Our methods are entirely algebraic and easily extend to Ore matrix polynomials having rational function coefficients.

The simultaneously row and column reduced form produced by these algorithms allows for both conversion to a first-order system of ordinary differential equations and the extraction of algebraic constraints. We have presented two algorithms for the computation of a simultaneous row and column reduced form. The first, given in Section 6.3.1 of Chapter 6, does a series of row-reduction and column-reduction as described in Section 6.2 of Chapter 6. However, a row-reduction followed by a column-reduction can undo a row-reduction, and similarly for a column-reduction followed by a row-reduction. Thus, while this approach is conceptually simpler, it has the drawback that we were unable so far to determine a polynomial complexity for the algorithm. It would be interesting in the future to have a closer and deeper analysis of this algorithm in order to see whether it has polynomial complexity. Our second method, algorithm **SRCR_Form** given in Section 6.5, has polynomial time complexity and is inspired by techniques used for computing Popov forms. Nevertheless, the order of the output system can in fact be higher than that of our first method, as we have seen in Example 6.4.5 of Chapter 6. Next task to do is the implementation and comparison of these algorithms.

All our algorithms are defined on power series coefficients since we want to use them for the analysis of the singularities of the systems. At present, the only methods for characterizing singularities are developed for first-order systems of ordinary differential equations. Ultimately, our goal is to be able to do a local analysis for higher-order differential systems and differential-algebraic equations directly without the need for conversion to first-order ordinary differential case. In order to accomplish this goal, we will need a generalization of the concept of Moser's reduction [77] and of super-reduction [58, 25] to higher-order differential systems. At present, this remains an open problem. We expect that the first step in such a direction would be a generalization of the Moser-reduction to linear differential-algebraic equations of first-order.

Résumé de la thèse

Dans cette thèse, nous considérons des équations différentielles matricielles de la forme

$$\mathcal{A}_\ell(x)\, y^{(\ell)}(x) + \mathcal{A}_{\ell-1}(x)\, y^{(\ell-1)}(x) + \cdots + \mathcal{A}_0(x)\, y(x) = f(x), \tag{E.1}$$

où ℓ est un entier naturel non nul, x est une variable complexe, les $\mathcal{A}_i$ sont des matrices de fonctions analytiques de taille $m \times n$, f est un vecteur de fonctions analytiques de dimension m, y est le vecteur d'inconnues de dimension n et $y^{(i)}(x) = \frac{d^i y}{dx^i}(x)$.

En fonction de la matrice de tête $\mathcal{A}_\ell(x)$, on distingue deux classes :

- si $\mathcal{A}_\ell(x)$ est une matrice carrée inversible, on parle alors de *systèmes différentiels explicites* ou de *systèmes d'équations différentielles ordinaires (ÉDO) linéaires*.
- si $\mathcal{A}_\ell(x)$ est une matrice carrée non inversible ou une matrice rectangulaire, on parle alors de *systèmes différentiels implicites* ou de *systèmes d'équations algébro-différentielles (ÉAD) linéaires*.

Les systèmes différentiels linéaires interviennent dans de nombreux problèmes en mathématiques : intégrabilité des systèmes dynamiques et hamiltoniens [6, 30], théorie de Galois différentielle (factorisation d'opérateurs différentiels scalaires [91, 98, 99] et résolution d'équations différentielles à coefficients dans des extensions liouvilliennes [43]) et etc. Ils apparaissent aussi dans la modélisation mathématique de nombreux problèmes issus de la chimie, la physique, la mécanique des fluides et la théorie de contrôle [34, 71, 73, 81, 82, 89]. L'exemple suivant provient d'un problème de la mécanique des fluides.

Exemple 1. *Les équations différentielles (2.17)-(2.19) de [71] donnent naissance au système différentiel linéaire suivant :*

$$\begin{pmatrix} 0 & 0 & 0 \\ -r^2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} y^{(3)} + \begin{pmatrix} m\,r & 0 & 0 \\ r & 0 & 0 \\ 0 & 0 & r \end{pmatrix} y'' + \begin{pmatrix} -m & -r^2 & 0 \\ m^2\,r^2 - 1 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} y' + \begin{pmatrix} -m^3\,r & 0 & 0 \\ 0 & m\,r^3 & \mathrm{Ra}\,r^3 \\ 0 & 0 & -m^2\,r \end{pmatrix} y = 0, \tag{E.2}$$

où r est la variable indépendante, m et Ra sont 2 constantes du problème et $y = \begin{pmatrix} \psi & \overline{p} & \overline{T} \end{pmatrix}^T$ est le vecteur d'inconnues. Comme la matrice de tête est non inversible, on a ainsi un système d'équations algébro-différentielles linéaires d'ordre 3. Pour résoudre ce système, les auteurs de [71] le transforment, par des éliminations différentielles, en une équation différentielle scalaire linéaire d'ordre 6, puis appliquent la méthode de Frobenius [44, 94] pour résoudre cette équation. ■

Cependant, il est intéressant du point de vue théorique et pratique de résoudre directement les systèmes (E.1) sans les réduire en une équation différentielle scalaire. Ainsi l'objectif principal de cette thèse est le développement de méthodes traitant directement les systèmes différentiels linéaires (E.1). Nous étudions ces systèmes théoriquement et algorithmiquement : nous implémentons des algorithmes en Maple et nous faisons une analyse de leur complexité arithmétique. Nous nous intéressons plus particulièrement à l'analyse locale de ces systèmes au voisinage d'un point x_0 qui n'est pas une singularité essentielle pour les matrices $\mathcal{A}_i$ et le vecteur f. Ainsi, nous supposons par la suite et sans aucune perte de généralité que $x_0 = 0$, $\mathcal{A}_i(x) \in \mathbb{K}[[x]]^{m\times n}$ pour $i = 0, \ldots, \ell$ et $f(x) \in \mathbb{K}[[x]]^m$, où $\mathbb{K}$ est un sous-corps du corps $\mathbb{C}$ des nombres complexes. Pour les systèmes d'ÉDO linéaires, nous nous intéressons à la détermination de la nature de la singularité (régulière ou irrégulière), au calcul des solutions formelles et etc. Quant aux systèmes d'ÉAD linéaires dont l'étude est en général plus compliquée que celle des

systèmes d'ÉDO linéaires, nous proposons de les simplifier en vue de les résoudre.

Les systèmes d'équations différentielles ordinaires linéaires du premier ordre, généralement donnés sous la forme

$$\frac{dy}{dx}(x) = A(x)\,y(x) + b(x),$$

où $A(x) \in \mathbb{K}((x))^{n\times n}$ et $b(x) \in \mathbb{K}((x))^n$, ont été initialement étudiés théoriquement pour démontrer l'existence de solutions formelles et déterminer leurs structures (voir [9] et ses références). Récemment, les recherches se sont tournées vers le calcul efficace de ces solutions [7, 58, 56, 13, 21, 85, 25]. De nombreux algorithmes ont vu le jour et ont été implémentés dans des systèmes de Calcul Formel, comme Maple, Mathematica et Singular.

En ce qui concerne les systèmes d'équations différentielles ordinaires linéaires d'ordre supérieur (E.1), l'approche classique consiste à les convertir en un système du premier ordre de la forme

$$\frac{d}{dx}\begin{pmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(\ell-2)}(x) \\ y^{(\ell-1)}(x) \end{pmatrix} = \begin{pmatrix} 0 & I_n & 0 & \cdots & 0 \\ 0 & 0 & I_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_n \\ \widetilde{\mathcal{A}}_0(x) & \widetilde{\mathcal{A}}_1(x) & \widetilde{\mathcal{A}}_2(x) & \ldots & \widetilde{\mathcal{A}}_{\ell-1}(x) \end{pmatrix} \begin{pmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(\ell-2)}(x) \\ y^{(\ell-1)}(x) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ f(x) \end{pmatrix},$$

où pour $i = 0, \ldots, \ell$, $\widetilde{\mathcal{A}}_i(x) = -\mathcal{A}_\ell^{-1}(x)\,\mathcal{A}_i(x)$, puis à appliquer les résultats et les algorithmes élaborés pour les systèmes du premier ordre (voir par exemple [29, 66]). Mais, l'inconvénient essentiel de cette approche est l'augmentation de la taille du problème ; pour un système différentiel de taille n et d'ordre ℓ, le système du premier ordre résultant est de taille $n\,\ell$. Cependant, on trouve dans la littérature une autre approche pour traiter ces systèmes, développée par Abramov, Bronstein & Khmelnov (voir [1, 2, 4, 5]). Dans ces travaux, les systèmes (E.1) sont transformés en des équations aux récurrences matricielles. Mais, cette approche est restreinte au cas des systèmes différentiels à coefficients polynomiaux. On trouve également dans la littérature quelques travaux directs comme par exemple dans [64, 79, 80], mais ceux-ci ne traitent que des cas particuliers de systèmes (E.1). Ainsi, l'absence de méthodes abordant directement tout système différentiel explicite (E.1) est la motivation majeure de cette thèse.

Considérons maintenant un système d'ÉAD linéaires du premier ordre de la forme

$$A(x)\,y'(x) + B(x)\,y(x) = f(x) \tag{E.3}$$

avec $A(x)$ une matrice rectangulaire ou une matrice carrée non inversible. Sans aucune perte de généralité, nous pouvons supposer que $A(x)$ est de la forme

$$A(x) = \begin{pmatrix} A_{11}(x) & 0 \\ 0 & 0 \end{pmatrix},$$

où $A_{11}(x)$ est une matrice carrée inversible. Décomposons la matrice $B(x)$ et les deux vecteurs $y(x)$ et $f(x)$ sous la forme

$$B(x) = \begin{pmatrix} B_{11}(x) & B_{12}(x) \\ B_{21}(x) & B_{22}(x) \end{pmatrix}, \quad y(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad \text{et} \quad f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}.$$

Le système (E.3) peut ainsi être écrit

$$\begin{cases} A_{11}(x)\,y_1'(x) + B_{11}(x)\,y_1(x) + B_{12}(x)\,y_2(x) = f_1(x), \\ B_{21}(x)\,y_1(x) + B_{22}(x)\,y_2(x) = f_2(x), \end{cases} \tag{E.4}$$

i.e., comme un ensemble composé d'équations algébriques et d'équations différentielles, d'où l'appellation systèmes d'équations algébro-différentielles. Cette classe de systèmes a été étudiée du point de vue numérique (voir [46, 67, 88]), où l'étude est divisée en deux parties essentielles : tout d'abord la recherche de valeurs initiales consistantes, puis le calcul des solutions. Comme la plupart des solveurs numériques nécessitent des équations différentielles ordinaires, on essaye alors d'extraire du système (E.3) une équation dite *équation différentielle sous-jacente*. Cette dernière est un système différentiel explicite linéaire obtenu en dérivant successivement les équations algébriques de (E.4) puis en faisant des manipulations algébriques pour exprimer y' en fonction de y et de x. Le nombre de dérivations nécessaires pour l'obtention de l'équation sous-jacente est appelé *indice différentiel* [46]. Il existe dans la littérature autres notions d'indices, voir [67, 89, 90].

Quant aux systèmes d'ÉAD linéaires d'ordre supérieur, comme dans la théorie classique des systèmes d'ÉDO linéaires, la méthode standard pour traiter ces systèmes consiste à les transformer en un système d'ÉAD du premier ordre puis résoudre ce dernier. Mais contrairement aux systèmes d'ÉDO, cette transformation en un système du premier ordre doit être effectuée avec une grande délicatesse car elle peut conduire à des difficultés mathématiques et numériques très importantes (voir [73]).

Du point de vue symbolique, on regarde les systèmes d'ÉAD linéaires en tant qu'opérateurs différentiels matriciels appliqués au vecteur inconnu $y(x)$ et on développe des algorithmes qui agissent sur ces opérateurs pour les transformer en une forme plus simple, telle la forme normale de Popov [40] (cette forme est utile pour exprimer les termes d'ordre supérieur en fonction des termes d'ordre plus bas et par suite transformer une ÉAD d'ordre supérieur en une ÉAD du premier ordre), la forme normale de Hermite [49] (une matrice triangulaire supérieure dont l'ordre est en général plus grand que l'ordre de l'opérateur initial) et la forme normale de Jacobson [38, 70, 74] (une matrice diagonale qui réduit le système différentiel en une équation différentielle scalaire linéaire). Une implémentation des algorithmes calculant ces formes normales est disponible dans Singular [51]. Dans cette thèse, nous nous intéressons aux algorithmes qui découplent un système donné d'ÉAD linéaires en un système purement différentiel et un autre purement algébrique.

Contenu de la thèse

Cette thèse est divisée en deux parties essentielles. Dans la première partie, qui est formée des chapitres 1 à 4, nous proposons des méthodes directes pour calculer les *solutions régulières formelles* des systèmes de la forme (E.1) avec $m = n$. Nous décrivons aussi une approche pour le calcul direct des formes dites *k-simples* ($k \in \mathbb{N}$) [14, 21, 85] des systèmes d'ÉDO linéaires du premier ordre. Ces formes sont utiles pour le calcul des solutions formelles sans ramification. La seconde partie, qui est composée des chapitres 5 et 6, porte sur les algorithmes de réduction des équations algébro-différentielles linéaires. Nous proposons des algorithmes séparant la composante algébrique de la composante différentielle.

En plus des résultats théoriques développés dans cette thèse, une autre contribution apparaît dans l'implémentation en Maple de la majorité des algorithmes élaborés (les programmes sont disponibles à `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`) et dans l'étude de leur complexité arithmétique. Finallement, il est important de noter que les algorithmes proposés dans la première partie de cette thèse peuvent être généralisés pour manipuler les systèmes aux différences et aux q-différences linéaires.

Le matériel de cette thèse est organisé comme suit. Le chapitre 1 contient un rappel des résultats classiques sur les matrices polynomiales [50, 65, 42, 104] que nous avons été amenés

à utiliser dans les chapitres suivants. Ce chapitre ne présente aucun contribution théorique, cependant nous avons étudié la complexité arithmétique et implémenté les algorithmes que nous utilisons dans la suite. Les cinq autres chapitres contiennent nos contributions. Nous résumons ci-dessous le contenu de chacun.

Chapitre 2 : Solutions régulières des systèmes différentiels linéaires d'ordre arbitraire de première espèce

Considérons un système d'ÉDO linéaires homogènes de la forme (E.1). Une base de son espace de solutions formelles est formée de $n\,\ell$ solutions de la forme

$$y_i(x) = \exp(q_i(x^{-1/r_i}))\, x^{\lambda_i}\, z_i(x^{1/r_i}), \qquad i = 1, \ldots, n\,\ell$$

où $r_i \in \mathbb{N}^*$, $q_i(t) \in t\overline{\mathbb{K}}[t]$, $\lambda_i \in \overline{\mathbb{K}}$ et $z_i(x^{1/r_i}) \in \overline{\mathbb{K}}[[x^{1/r_i}]][\ln(x)]^n$. Une *solution régulière formelle* de (E.1) est une combinaison linéaire de solutions de la forme

$$y(x) = x^{\lambda_0}\, z(x), \tag{E.4}$$

où $\lambda_0 \in \overline{\mathbb{K}}$ et $z(x) \in \mathbb{K}[[x]][\ln(x)]^n$. Le point $x = 0$ est une singularité régulière de (E.1) si ce dernier admet $n\,\ell$ solutions régulières linéairement indépendantes. Dans ce chapitre et le suivant, nous abordons le problème du calcul des solutions régulières formelles d'un système de la forme (E.1). Plus particulièrement, nous nous intéressons au calcul d'une base de l'espace des solutions régulières formelles formée de solutions de la forme (E.4).

Pour des raisons de commodité, nous allons écrire le système différentiel considéré avec la dérivation d'Euler $\vartheta = x\,\frac{d}{dx}$. Ainsi, nous allons traiter un système de la forme

$$\mathcal{L}(x, \vartheta)(y(x)) = A_\ell(x)\,\vartheta^\ell(y(x)) + A_{\ell-1}(x)\,\vartheta^{\ell-1}(y(x)) + \cdots + A_0(x)\,y(x) = 0, \tag{E.5}$$

où pour $i = 0, \ldots, \ell$, $A_i(x) \in \mathbb{K}[[x]]^{n\times n}$ et $A_\ell(x)$ est inversible.

Dans ce chapitre, nous supposons que $A_\ell(x)$ est inversible en $x = 0$, et nous faisons référence au système (E.5) comme un *système de première espèce*. Dans ce cas, la singularité $x = 0$ est une singularité régulière pour le système.

Lorsque $n = 1$ (une seule équation différentielle), dire que $A_\ell(0)$ est inversible est équivalent à dire que $A_\ell(0)$ est non nul. Dans ce cas, l'exposant λ_0 de toute solution régulière de la forme (E.4) est choisi comme une racine d'un polynôme appelé *polynôme indiciel*. Parmi les méthodes calculant les solutions régulières formelles d'une équation différentielle ordinaire scalaire, nous citons la méthode de Frobenius [44, 37] et ses deux variantes, celle de Heffter [53] et celle de Poole [86].

Dans ce chapitre, nous proposons deux méthodes pour calculer les solutions régulières formelles d'un système de première espèce de la forme (E.5). La première méthode est inspirée par l'approche de Poole [86] dans le cas scalaire. L'idée à la base de cette méthode est de regarder les solutions régulières (E.4) comme des séries en x dont les coefficients sont polynomiaux en $t = \ln(x)$ (et donc $\vartheta = \frac{d}{dt}$), c'est-à-dire, de la forme

$$y(x) = x^{\lambda_0}(U_0(t) + U_1(t)\,x + \cdots + U_i(t)\,x^i + \cdots)$$

avec $\lambda_0 \in \overline{\mathbb{K}}$, pour $m \geq 0$, $U_m(t) \in \overline{\mathbb{K}}[t]^n$ et $U_0 \neq 0$. Écrivons le système (E.5) en regroupant ses termes suivant les puissances de x, c'est-à-dire,

$$\mathcal{L}(x, \vartheta)(y(x)) = \sum_{j=0}^{\infty} x^j\, L_j(\vartheta)(y(x)) = 0,$$

où pour $j \geq 0$,

$$L_j(\lambda) = A_{\ell,j}\,\lambda^\ell + \cdots + A_{1,j}\,\lambda + A_{0,j},$$

$A_{i,j}$ étant le coefficient de x^j du développement de Taylor de $A_i(x)$.

En injectant $y(x)$ dans (E.5), nous trouvons

$$L_0(\vartheta)\left(x^{\lambda_0}\,U_0\right) = 0, \tag{E.6}$$

et

$$\forall\, i \geq 1, \quad L_0(\vartheta + \lambda_0 + i)(U_i) = -\sum_{j=0}^{i-1} L_{i-j}(\vartheta + \lambda_0 + j)(U_j). \tag{E.7}$$

Proposition 1. *Le vecteur $x^{\lambda_0}\,U_0$ est une solution régulière du système $L_0(\vartheta)(y(x)) = 0$ si et seulement si λ_0 est une valeur propre de la matrice polynomiale $L_0(\lambda)$, c'est-à-dire,* $\det(L_0(\lambda_0)) = 0$*, et U_0 est de la forme*

$$U_0 = \sum_{i=0}^{k-1} v_{k-1-i}\frac{\ln^i(x)}{i!},$$

où $0 \neq v_0, \ldots, v_{k-1} \in \overline{\mathbb{K}}^n$ forment une chaîne de Jordan pour $L_0(\lambda)$ associée à la valeur propre λ_0, c'est-à-dire, satisfont

$$\sum_{j=0}^{i} \frac{1}{j!} L_0^{(j)}(\lambda_0)\, v_{i-j} = 0, \quad \textit{for } i = 0, \ldots, k-1.$$

Nous remarquons ainsi que le déterminant de $L_0(\lambda)$ joue le même rôle que le polynôme indiciel dans le cas scalaire. Notons que, comme nous considérons un système de première espèce, *i.e.*, la matrice $A_\ell(0)$ est supposée inversible, le degré du déterminant de $L_0(\lambda)$ est égal à $n\,\ell$.

Une fois que le choix de λ_0 et U_0 est fait et que les U_j pour $j = 1, \ldots, i-1$ sont calculés (ceux sont des vecteurs de polynômes en $\ln(x)$), le coefficient U_i satisfait alors un système différentiel linéaire non-homogène à coefficients constants dont le second membre est un vecteur de polynômes en $\ln(x)$. La proposition suivante montre l'existence d'une solution polynomiale en $\ln(x)$ de ce dernier système et donne une borne sur son degré.

Proposition 2. *Le système (E.7) admet au moins une solution polynomiale en* $\ln(x)$ *de degré p avec*

$$d \leq p \leq d + \max_j\{\kappa_j\},$$

où d désigne le degré en $\ln(x)$ *du second membre de (E.7) et*

$$\kappa_j = \begin{cases} \textit{multiplicité partielle de } \lambda_0 + i, & \textit{si } \lambda_0 + i \text{ est une valeur propre de } L_0(\lambda), \\ 0, & \textit{sinon.} \end{cases}$$

Pour résoudre les systèmes (E.6) et (E.7), nous réduisons le problème à la résolution de systèmes algébriques linéaires ayant une structure par bloc très particulière. Nous obtenons ainsi un algorithme qui, pour un système de première espèce donné de la forme (E.5), retourne une base de son espace de solutions formelles. Nous donnons deux variantes de cet algorithme. Dans la première variante, chaque solution $x^{\lambda_0}U_0$ d'un système fondamental de solutions du système $L_0(\vartheta)(y(x)) = 0$ est complétée pour former une solution du système (E.5). Dans la seconde variante, nous regroupons les valeurs propres de la matrice polynomiale $L_0(\lambda)$ en des ensembles disjoints de sorte que les valeurs propres appartenant à deux ensembles distincts ne diffèrent pas d'entiers. Puis, pour chaque ensemble, nous calculons la solution régulière générale de (E.5) générée par ces valeurs propres. Nous étudions la complexité arithmétique de cet algorithme:

Proposition 3. *Notre algorithme calcule la solution régulière générale jusqu'à l'ordre ν d'un système de première espèce de la forme (E.5) en au plus $O\big(n^4\,\ell^3\,\nu^2 + n^6\,\ell^4\big)$ opérations dans $\mathbb{K}$.*

Nous présentons aussi quelques tables de temps de calculs comparant l'implémentation en MAPLE de notre méthode (disponible à `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`) avec deux autres méthodes : celle présentée dans [5] qui réduit le problème du calcul des solutions régulières des systèmes de la forme (E.5) à coefficients polynomiaux $A_i(x)$ à celui du calcul des solutions sous forme de séries de Laurent pour des équations aux récurrences matricielles (implémentée dans le package LINEARFUNCTIONALSYSTEMS de MAPLE), et celle qui consiste à convertir (E.5) en système du premier ordre de taille $n\,\ell$ puis appliquer l'algorithme de [21] (implémentée dans le package ISOLDE de MAPLE). Les tableaux montrent l'efficacité de notre algorithme spécialement sur des systèmes d'ordre $\ell \geq 2$.

La deuxième méthode que nous proposons dans ce chapitre pour calculer les solutions régulières d'un système de première espèce de la forme (E.5) est une généralisation de la méthode de Frobenius [44, 94]. Celle-ci a été généralisée aux systèmes du premier ordre dans [92, 54, 56] et aux systèmes d'ordre supérieur dans [64, 80, 81], mais les travaux (sur l'ordre supérieur) de ces derniers papiers ne traitent pas le cas général. Notre généralisation suit les grands traits de la méthode dans le cas scalaire et utilisent des résultats classiques sur les matrices polynomiales [50] (notamment les notions de multiplicités partielles et de racines polynomiales) ce qui la distinguent des autres généralisations existantes.

Ce chapitre est un travail en collaboration avec M. A. Barkatou et T. Cluzeau et a été publié dans [16] et une partie de [17, 18].

Chapitre 3 : Formes simples et applications au calcul des solutions régulières

Ce chapitre est le sujet de l'article publié [18] en collaboration avec M. A. Barkatou et T. Cluzeau.

Nous avons vu au chapitre précédent comment calculer les solutions régulières des systèmes de première espèce. Dans ce chapitre, nous montrons comment calculer les solutions régulières d'une classe de systèmes plus générale.

Nous considérons un système de la même forme que (E.5) mais nous supposons désormais que $A_\ell(x)$ et $A_\ell(0)$ sont deux matrices quelconques (pas forcément inversibles). À un tel système, nous associons la matrice polynomiale

$$L_0(\lambda) = A_\ell(0)\,\lambda^\ell + \cdots + A_1(0)\,\lambda + A_0(0). \tag{E.8}$$

Nous examinons tout d'abord le cas où la matrice $L_0(\lambda)$ est régulière, *i.e.*, $\det(L_0(\lambda)) \neq 0$.

Définition 1. *Un système de la forme (E.5), respectivement l'opérateur associé $\mathcal{L}(x, \vartheta)$, ayant une matrice polynomiale $L_0(\lambda)$ régulière est appelé* système simple, *respectivement* opérateur simple.

La notion de systèmes simples a été introduite par Barkatou dans [14] pour calculer les solutions rationnelles des systèmes d'ÉDO linéaires du premier ordre à coefficients rationnels. Puis, elle a été utilisée par Barkatou & Pflügel [21] dans le calcul des solutions régulières (toujours pour des systèmes du premier ordre). La terminologie systèmes simples se justifie par le fait que les polynômes indiciels de tels systèmes peuvent être facilement calculés.

Remarquons qu'un système de première espèce est nécessairement simple mais l'inverse n'est pas toujours vraie. Soulignons aussi que la classe des systèmes simples peut contenir des systèmes d'ÉAD, vu que la matrice de tête $A_\ell(x)$ n'est pas supposée inversible.

Nous montrons dans ce chapitre que les méthodes développées dans le chapitre 2 pour calculer les solutions régulières formelles d'un système de première espèce restent valables pour les systèmes simples et que :

Théorème 1. *La dimension de l'espace des solutions régulières formelles d'un système différentiel linéaire simple de la forme (E.5) est égale au degré du déterminant de la matrice polynomiale $L_0(\lambda)$ donnée par (E.8).*

Le théorème ci-dessus nous permet de déterminer la nature de la singularité $x = 0$ pour des systèmes différentiels explicites simples :

Corollaire 1. *Un système différentiel linéaire simple de la forme (E.5) ayant une matrice de tête $A_\ell(x)$ inversible dans $\mathbb{K}((x))^{n\times n}$ a une singularité régulière en $x = 0$ si et seulement si le système est de première espèce.*

Une autre contribution de ce chapitre est l'étude de la complexité arithmétique de l'algorithme développé dans [21] calculant les solutions régulières d'un système simple du premier ordre. Ceci donne $O\big(n^4\,\nu^2 + n^5\,\nu\big)$ opérations dans $\mathbb{K}$, où n désigne la taille du système et ν désigne l'ordre auquel les solutions sont calculées. Le but de cette étude est de comparer, du point de vue complexité, notre approche (la généralisation de la méthode de Poole) pour résoudre directement un système simple de la forme (E.5) et celle qui consiste à transformer le système (E.5) en un autre système du premier ordre de taille $n\,\ell$ puis utiliser l'algorithme de [21]. Nous constatons que notre approche a une meilleur complexité quand ℓ est largement plus grande que n.

Puis, nous examinons le problème du calcul des solutions régulières pour les systèmes non simples, *i.e.*, les systèmes de la forme (E.5) dont la matrice polynomiale $L_0(\lambda)$ est singulière ($\det(L_0(\lambda)) = 0$). Comme les algorithmes décrits au chapitre 2 ne s'appliquent pas à ces systèmes, nous proposons alors de calculer un système différentiel linéaire simple $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ à partir duquel nous pouvons déduire les solutions régulières du système non simple. Afin de garantir l'existence d'un tel système $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$, nous supposons que la matrice de tête $A_\ell(x)$ du système (E.5) est inversible dans $\mathbb{K}((x))^{n\times n}$ (*i.e.*, nous ne considérons que des systèmes différentiels explicites). Ceci implique que l'espace des solutions régulières formelles de (E.5) est de dimension finie.

Le problème du calcul d'un système simple à partir duquel on peut retrouver les solutions du système non simple $\mathcal{L}(x,\vartheta)(y(x)) = 0$ a déjà été étudié dans [21] (voir aussi [14]) pour le cas des systèmes du premier ordre ($\ell = 1$) ; il a été montré qu'en utilisant l'algorithme de la super-réduction [24, 25, 58], on peut construire deux matrices $S(x)$ et $T(x)$ inversibles dans $\mathbb{K}((x))^{n\times n}$ telles que l'opérateur $\overline{\mathcal{L}}(x,\vartheta) = S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ est simple. Par conséquent, l'espace des solutions formelles de $\mathcal{L}(x,\vartheta)(y(x)) = 0$ et celui de $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$ sont isomorphes vu que $y(x)$ et $z(x)$ sont reliés par $y(x) = T(x)\,z(x)$ avec $T(x)$ inversible.

En passant aux systèmes non simples d'ordre supérieur ($\ell \geq 2$), il n'est malheureusement pas toujours possible de trouver deux matrices inversibles $S(x)$ et $T(x)$ telles que l'opérateur $S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ est simple. En effet, nous montrons que, étant donné un système non simple de la forme (E.5) avec $n = \ell = 2$, si une base minimale à droite et une base minimale à gauche [65, 42] de la matrice polynomiale $L_0(\lambda)$ donnée par (E.8) sont formées uniquement de vecteurs non constants, alors pour toutes matrices $S(x)$ et $T(x)$ dans $\mathbb{K}((x))^{2\times 2}$, l'opérateur $S(x)\,\mathcal{L}(x,\vartheta)\,T(x)$ est toujours non simple. Pour cette raison, nous nous intéressons tout d'abord à l'existence d'une substitution linéaire $y(x) = T(x)\,z(x)$, où $T(x)$ est une matrice inversible,

telle que le nouveau système différentiel satisfait par $z(x)$ est simple. Nous donnons une condition nécessaire sur l'existence d'une telle substitution linéaire :

Théorème 2. *Étant donné un système non simple d'ÉDO linéaires de la forme (E.5), s'il existe une matrice inversible $T(x) \in \mathbb{K}((x))^{n\times n}$ telle que le système $\overline{\mathcal{L}}(x,\vartheta)(z(x)) = 0$, où $\overline{\mathcal{L}}(x,\vartheta) = \mathcal{L}(x,\vartheta)\,T(x)$, est simple, alors les éléments d'une base minimale à droite de la matrice polynomiale $L_0(\lambda)$ donnée par (E.8) sont tous des vecteurs constants.*

En nous basant sur le théorème ci-dessus, nous développons un algorithme qui, soit décide de l'existence d'une telle substitution linéaire et la calcule, soit montre qu'elle n'existe pas. Dans ce dernier cas, nous proposons une variante différentielle de l'algorithme *EG'* proposé par Abramov, Bronstein & Khmelnov dans [4, Section 4]. Ce dernier algorithme qui est une amélioration de l'algorithme *EG* décrit dans [1] ramène une équation aux récurrences matricielle à une autre équation ayant sa matrice de tête, ou de queue, inversible. Dans ce chapitre, nous adaptons cet algorithme à notre cas : nous partons d'un système différentiel linéaire ayant une matrice polynomiale associée $L_0(\lambda)$ singulière et nous voulons calculer un autre système dont la matrice polynomiale associée est régulière. Pour cela, nous avons besoin de supposer que le système non simple $\mathcal{L}(x,\vartheta)(y(x)) = 0$ a des coefficients polynomiaux. L'algorithme applique une série d'opérations élémentaires sur les lignes de l'opérateur $\mathcal{L}(x,\vartheta)$ et retourne un système simple à partir duquel les solutions régulières de $\mathcal{L}(x,\vartheta)(y(x)) = 0$ peuvent être déduites. Notons que le système simple obtenu pourrait être d'ordre plus grand que celui de $\mathcal{L}(x,\vartheta)(y(x)) = 0$ et qu'il n'est pas forcément équivalent à $\mathcal{L}(x,\vartheta)(y(x)) = 0$ dans le sens où les espaces de solutions formelles des deux systèmes peuvent ne pas être isomorphes ; quand ils ne le sont pas, nous expliquons comment les solutions régulières du premier système peuvent être obtenues.

Une autre contribution de ce chapitre se présente par l'étude de la complexité arithmétique des nouveaux algorithmes développés et leur implémentation en Maple.

Chapitre 4 : Sur les formes k-simples des systèmes différentiels linéaires du premier ordre et leur calcul

L'idée à l'origine de ce chapitre provient du travail [85] de Pflügel.
Nous considérons un système d'ÉDO linéaires du premier ordre de la forme

$$\vartheta(y(x)) = A(x)\,y(x) \quad \text{avec } A(x) = \frac{1}{x^p}\left(A_0 + A_1\,x + \cdots + A_k\,x^k + \cdots\right), \tag{E.9}$$

où $\vartheta = x\frac{d}{dx}$, $p \in \mathbb{N}$ et les A_i sont des matrices de taille $n \times n$ dont les entrées sont dans $\mathbb{K}$ telles que $A_0 \neq 0$. Dans [85], afin de calculer les solutions formelles de (E.9), l'auteur est amené à considérer des systèmes différentiels sous une forme un peu plus générale que (E.9). En effet, pour un entier $k \in \{0,\ldots,p-1\}$, on définit la matrice $D(x) = \operatorname{diag}(x^{\alpha_1},\ldots,x^{\alpha_n})$ avec $\alpha_i = \max\{0, -k - v(A(x)(i,.))\}$, où $A(x)(i,.)$ désigne la $i$ème ligne de la matrice $A(x)$. En multipliant le système (E.9) à gauche par $x^k D(x)$, on obtient

$$\mathcal{D}_k(y(x)) = D(x)\,\vartheta_k(y(x)) + N(x)y(x) = 0, \tag{E.10}$$

où $\vartheta_k = x^k\,\vartheta$ et $N(x) = -x^k D(x)A(x) \in \mathbb{K}[[x]]^{n\times n}$.

En étudiant l'action de l'opérateur $\mathcal{D}_k$ sur une solution de la forme $y = \exp\left(\int w\right) z$ avec

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots \in \overline{\mathbb{K}((x))},$$

et $z \in \overline{\mathbb{K}}[[x^{\frac{1}{r}}]]^n$ $(r \in \mathbb{N}^*)$ tel que $z(0) \neq 0$, on trouve

$$\exp\Big(-\int w\Big)\,\mathcal{D}_k(y) = (D(0)\,\lambda_0 + N(0))\,z(0) + \cdots,$$

où ... représente des termes de valuation plus grande. Il en découle que si $y = \exp\big(\int w\big)\,z$ est une solution of $\mathcal{D}_k(y) = 0$, alors on a $(D(0)\,\lambda_0 + N(0))\,z(0) = 0$ ce qui donne les deux conditions suivantes :

$$\det(D(0)\,\lambda_0 + N(0)) = 0 \quad \text{et} \quad z(0) \in \ker(D(0)\,\lambda_0 + N(0)).$$

Il est donc naturel par la suite de s'attendre à ce que les racines du déterminant du faisceau

$$L_k(\lambda) = D(0)\,\lambda + N(0)$$

jouent un rôle important dans la détermination des solutions formelles. Mais, il se peut bien évidemment que le déterminant $\det(L_k(\lambda))$ soit identiquement nul auquel cas on ne peut tirer aucune information utile.

Un système de la forme (E.10) ayant un faisceau $L_k(\lambda)$ régulier est appelé *système k-simple* ou *système simple par rapport à ϑ_k*. Ainsi, la notion de k-simplicité vient comme une généralisation de la notion de simplicité vue dans le chapitre précédent et dans [14, 21].

Il a été montré dans [85, Th. 3.3] que si le système (E.9) peut être écrit comme un système k-simple de la forme (E.10) et si λ_0 est une valeur propre du faisceau $L_k(\lambda)$ de multiplicité algébrique m, alors il existe m solutions formelles linéairement indépendantes de (E.9) de la forme $y = \exp\big(\int w\big)\,z$ avec

$$w = \frac{\lambda_0}{x^{k+1}} + \cdots \in x^{-1}\overline{\mathbb{K}}[x^{-1/r}] \quad (r \in \mathbb{N}^*),$$

où ... représente des termes de valuation plus grande, et $z \in \overline{\mathbb{K}}[[x^{1/r}]]^n[\ln(x)]$. Par suite, le déterminant de $L_k(\lambda)$ joue le même rôle que les polynômes de Newton (ou les polynômes caractéristiques) dans le cas scalaire.

Notons qu'un système de la forme (E.9) écrit comme dans (E.10) n'est pas toujours k-simple. Cependant, il a été remarqué que les polynômes définis à partir d'une forme super-réduite [58, 24, 25] (les formes super-réduites ne sont définies que pour les systèmes de la forme (E.9)) sont fortement reliés aux polynômes de Newton. Pflügel a montré dans [85] que si le système (E.9) est super-réduit, alors il peut être écrit comme un système k-simple pour $k = 0, \ldots, p-1$.

Dans ce chapitre, nous considérons la classe des systèmes de la forme (E.10) avec $k \in \mathbb{N}$, $D(x)$ et $N(x)$ dans $\mathbb{K}[[x]]^{n\times n}$ telles que $D(x)$ est inversible dans $\mathbb{K}((x))^{n\times n}$ ($D(x)$ n'est pas forcément supposée diagonale). Avant cette thèse, l'unique méthode qui existait pour calculer un système k-simple équivalent à (E.10) (sans le réduire à une équation différentielle linéaire scalaire) était en utilisant l'algorithme de la super-réduction. Mais, un système k-simple de la forme (E.10), écrit sous la forme (E.9) (avec $A(x) = -x^{-k}\,D^{-1}(x)\,N(x)$), n'est pas nécessairement super-réduit. Dans ce chapitre, nous supposons que la matrice de tête $D(x)$ est de la forme $D(x) = x^{\alpha}Q(x)$ où $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ tel que $\alpha_1 \leq \cdots \leq \alpha_n$ et $Q(x) \in \mathbb{K}[[x]]^{n\times n}$ (on peut toujours ramener $D(x)$ à cette forme en multipliant l'opérateur $\mathcal{D}_k$ à gauche et à droite par deux matrices unimodulaires de $\mathbb{K}[[x]]^{n\times n}$) et nous montrons que :

Théorème 3. *Il existe deux matrices inversibles $U(x) \in \mathbb{K}[x^{-1}]^{n\times n}$ avec $U^{-1}(x) \in \mathbb{K}[x]^{n\times n}$ et $V(x) \in \mathbb{K}[x]^{n\times n}$ avec $V^{-1}(x) \in \mathbb{K}[x^{-1}]^{n\times n}$ telles que l'opérateur $U(x)\,\mathcal{D}_k\,V(x)$ est k-simple.*

Nous développons un algorithme, implémenté en MAPLE et disponible à `http://www.unilim.fr/pages_perso/carole.el-bacha/recherche.html`, qui retourne

deux matrices $U(x)$ et $V(x)$ satisfaisant les propriétés du Théorème 3 en au plus $O\big(n^{\omega+1}\,|\alpha(\mathcal{D}_k)| + \nu\, n^3\,|\alpha(\mathcal{D}_k)|\big)$ opérations dans $\mathbb{K}$, où $\alpha(\mathcal{D}_k) = \sum_{i=1}^{n} \alpha_i$ et ν est l'ordre de troncation des matrices $D(x)$ et $N(x)$. L'entier ν doit être choisi plus grand ou égal à $|\alpha(\mathcal{D}_k)|$. Nous illustrons notre algorithme par des exemples clarifiant ses différentes étapes.

Notre démarche pour calculer deux matrices $U(x)$ et $V(x)$ comme ci-dessus repose sur le traitement algébrique du faisceau $L_k(\lambda)$ et procède d'une manière semblable à celle utilisée dans [58].

Finalement, nous soulignons quelques caractéristiques de notre algorithme :

- Le système k-simple produit par notre algorithme n'est pas forcément super-réduit.
- Notre algorithme appliqué à $\mathcal{D}_k$ donné par (E.10) avec $k = 0$ nous permet de déterminer la nature de la singularité $x = 0$. Il peut être ainsi considéré comme une approche alternative à la réduction de Moser [77] pour déterminer la nature de la singularité $x = 0$.
- Notre approche appliquée à $\mathcal{D}_k$ donné par (E.10) préserve la simplicité par rapport à ϑ_{k+i} pour $i = 1, 2, \ldots$. En d'autres termes, si l'opérateur $\mathcal{D}_k$ écrit respectivement avec ϑ_{k+i} pour $i = 1, 2, \ldots$ est simple par rapport à ϑ_{k+i}, alors après avoir appliqué notre algorithme à $\mathcal{D}_k$, l'opérateur retourné est aussi simple par rapport à ϑ_{k+i} pour $i = 1, 2, \ldots$.

Chapitre 5 : Algorithmes de réduction pour les systèmes d'ÉAD linéaires du premier ordre

Le contenu de ce chapitre constitue une partie du papier publié [20] en collaboration avec M. A. Barkatou et E. Pflügel.

Nous étudions un système d'équations algébro-différentielles linéaires du premier ordre de la forme

$$L(y(x)) = A(x)\,\partial(y(x)) + B(x)\,y(x) = f(x), \tag{E.11}$$

où $\partial = \frac{d}{dx}$, $A(x)$ et $B(x)$ sont dans $\mathbb{K}[[x]]^{m\times n}$ et $f(x)$ dans $\mathbb{K}[[x]]^m$. Les questions qui nous intéressent sur les systèmes d'ÉAD sont liées à la consistance du système, aux types de solutions qu'ils admettent et à la manière de les calculer. Pour répondre à ces questions, une idée naturelle est de réduire le système (E.11) à une forme standard qui pourrait être manipulée par des techniques bien connues comme, par exemple, celles développées pour les systèmes algébriques ou pour les systèmes différentiels ordinaires linéaires.

Indépendamment de leurs objectifs, la plupart des algorithmes symboliques ou numériques manipulant les systèmes d'ÉAD linéaires du premier ordre (voir [47, 67, 88]) appliquent les mêmes techniques :

- multiplier le système à gauche par une matrice inversible,
- dériver ses équations algébriques,
- faire un changement de variable $y(x) = T(x)\,z(x)$ avec $T(x)$ une matrice inversible.

Les opérations ci-dessus ne suffisent pas toujours pour découpler entièrement un système de la forme (E.11) en un système d'ÉDO et un système d'équations algébriques. Cependant, en regardant de plus près l'algorithme de Harris, Sibuya & Weinberg [52], nous avons remarqué qu'en autorisant aussi des changements de variable de la forme $y(x) = T(z(x))$, où T est un opérateur différentiel matriciel unimodulaire, nous arrivons à découpler entièrement le système.

Dans ce chapitre, nous étudions tout d'abord l'algorithme de Harris et al. [52]. Nous le présentons d'une nouvelle façon en exprimant les opérations effectuées sur le système comme une série de transformations à gauche et à droite sur l'opérateur différentiel matriciel L. Puis, inspiré par ce dernier algorithme, nous donnons une nouvelle approche plus simple menant au théorème suivant :

Théorème 4. *Étant donné un opérateur différentiel matriciel $L = A(x)\,\partial + B(x) \in \mathbb{K}[[x]][\partial]^{m\times n}$, il existe deux opérateurs différentiels matriciels unimodulaires S et T tels que $\widetilde{L} = S\,L\,T$ est de la forme*

$$\widetilde{L} = \begin{pmatrix} \widetilde{A}_{11}(x)\,\partial + \widetilde{B}_{11}(x) & 0 & 0 \\ 0 & \widetilde{B}_{22}(x) & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{E.12}$$

où $\widetilde{A}_{11}$ et $\widetilde{B}_{22}$ sont des matrices inversibles.

Ainsi le système d'ÉAD (E.11) peut être découplé en 2 systèmes :

1. le système d'ÉDO linéaires du premier ordre $\widetilde{A}_{11}(x)\,z_1'(x) + \widetilde{B}_{11}(x)\,z_1(x) = \widetilde{f}_1(x)$,
2. le système d'équations algébriques $\widetilde{B}_{22}(x)\,z_2(x) = \widetilde{f}_2(x)$,

avec quelques conditions nécessaires sur le second membre exprimées par $\widetilde{f}_3(x) = 0$ pour assurer la consistance du système. Ici, on a

$$y(x) = T\begin{pmatrix}\begin{pmatrix} z_1(x) \\ z_2(x) \\ z_3(x)\end{pmatrix}\end{pmatrix} \quad \text{et} \quad \begin{pmatrix} \widetilde{f}_1(x) \\ \widetilde{f}_2(x) \\ \widetilde{f}_3(x)\end{pmatrix} = S(f(x)).$$

Pour calculer la forme découplée (E.12), nous procédons comme suit. Nous commençons par réduire l'opérateur différentiel matriciel L donné en entrée suivant les lignes en utilisant la procédure décrite dans [27]. Ceci se traduit par une multiplication de L à gauche par un opérateur différentiel matriciel unimodulaire convenable. Puis, nous suivons cette opération par une réduction suivant les colonnes de l'opérateur obtenu. L'opérateur résultant est donc réduit par rapport aux colonnes mais pas forcément par rapport aux lignes. S'il ne l'est pas, nous ré-appliquons la réduction suivant les lignes et ainsi de suite jusqu'à ce que nous trouvons un opérateur qui est simultanément réduit par rapport aux lignes et aux colonnes. Nous montrons que ceci se produira après un nombre fini d'itérations. Finalement, par multiplication à gauche et à droite par des matrices inversibles, nous obtenons un opérateur de la forme (E.12).

Chapitre 6 : Systèmes différentiels linéaires simultanément réduits par rapport aux lignes et aux colonnes

Ce chapitre fait l'objet de l'article [19] en collaboration avec M. A. Barkatou, G. Labahn et E. Pflügel.

Nous généralisons dans ce chapitre la méthode proposée dans le chapitre 5 (pour découpler une ÉAD du premier ordre) aux systèmes d'ÉAD linéaires d'ordre supérieur de la forme (E.1). Ainsi, nous agissons sur l'opérateur différentiel matriciel L définissant le système (E.1), *i.e.*,

$$L = \mathcal{A}_\ell(x)\,\partial^\ell + \mathcal{A}_{\ell-1}(x)\,\partial^{\ell-1} + \cdots + \mathcal{A}_0(x),$$

afin d'obtenir un autre opérateur différentiel matriciel qui lui est équivalent et qui est *simultanément réduit par rapport aux lignes et aux colonnes*. Ce dernier est d'ordre inférieur ou égal à celui de L et a une structure par blocs très particulière comme le montre la proposition suivante :

Proposition 4. *Un opérateur différentiel matriciel $L \in \mathbb{K}[[x]][\partial]^{m\times n}$ est simultanément réduit par rapport aux lignes et colonnes si et seulement si L a, à une permutation près, la partition par blocs suivante :*

$$\left(\begin{array}{ccc|c} L_{11} & \cdots & L_{1k} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ L_{k1} & \cdots & L_{kk} & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right),$$

où,

(a) pour $i = 1, \ldots, k$, les L_{ii} sont des opérateurs différentiels matriciels carrés ayant une matrice de tête inversible et satisfont $\operatorname{ord}(L_{ii}) > \operatorname{ord}(L_{i+1\,i+1})$,

(b) $\operatorname{ord}(L_{ij}) \leq \operatorname{ord}(L_{ii})$ *pour* $j < i$ *et* $\operatorname{ord}(L_{ij}) \leq \operatorname{ord}(L_{jj})$ *pour* $i < j$.

Construire un opérateur simultanément réduit par rapport aux lignes et aux colonnes a de conséquences avantageuses sur le système différentiel qui lui est associé. En effet, il permet de découpler ce dernier en un système purement algébrique, un système purement différentiel, *i.e.*, carré dont toutes les équations sont linéairement indépendantes, et quelques conditions nécessaires sur le second membre. De plus, le système purement différentiel obtenu peut être facilement réécrit comme un système différentiel explicite du premier ordre, et par suite, la dimension de son espace de solutions formelles peut être déterminée et la classification de la singularité $x = 0$ peut aussi être explorée.

Pour calculer une forme simultanément réduite par rapport aux lignes et aux colonnes, nous proposons deux approches. La première approche consiste à appliquer alternativement une série de réduction par lignes et réduction par colonnes jusqu'à trouver un opérateur qui est à la fois réduit par rapport aux lignes et aux colonnes. Bien que cette méthode agisse sur les lignes et les colonnes individuellement, l'opérateur obtenu a une structure par blocs. La deuxième approche est inspirée des techniques utilisées pour calculer la forme normale de Popov dans le cas commutatif (voir [78]) sauf que ici nous appliquons cette procédure à des blocs.

Finallement, nous notons qu'une forme simultanément réduite par rapport aux lignes et aux colonnes est plus faible que les formes normales de Popov et de Jacobson.

Nous concluons ce résumé par un exemple illustrant le contenu de cette thèse.

Exemple 2. *Nous considérons le système d'ÉAD linéaires donné par (E.2) et nous nous intéressons aux types de solutions que possède ce système. Tout d'abord, écrivons le sous la forme $L(y) = 0$ où L est l'opérateur différentiel matriciel donné par*

$$L = \begin{pmatrix} m\,r\,\partial^2 - m\,\partial - m^3\,r & -r^2\,\partial & 0 \\ -r^2\,\partial^3 + r\,\partial^2 + (m^2\,r^2 - 1)\,\partial & m\,r^3 & \mathrm{Ra}\,r^3 \\ -\partial & 0 & r\,\partial^2 + \partial - m^2\,r \end{pmatrix}.$$

Puis, utilisons l'algorithme développé au chapitre 6 pour rendre L simultanément réduit par rapport aux lignes et aux colonnes. Nous obtenons ainsi l'opérateur

$$\widetilde{L} = S\,L\,T = \begin{pmatrix} m\,r\,\partial^2 - m\,\partial - m^3\,r & -r^2\,\partial & 0 \\ m\,r\,\partial^2 - m\,\partial - m^3\,r & -r^3\,\partial^2 - 2\,r^2\,\partial + m^2\,r^3 & \mathrm{Ra}\,m\,r^3 \\ -\partial & 0 & r\,\partial^2 + \partial - m^2\,r \end{pmatrix},$$

avec $T = I_3$ et

$$S = \begin{pmatrix} 1 & 0 & 0 \\ r\,\partial & m & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Le système (E.2) est donc équivalent au système

$$\begin{pmatrix} m\,r & 0 & 0 \\ m\,r & -r^3 & 0 \\ 0 & 0 & r \end{pmatrix} y'' + \begin{pmatrix} -m & -r^2 & 0 \\ -m & -2\,r^2 & 0 \\ -1 & 0 & 1 \end{pmatrix} y' + \begin{pmatrix} -m^3\,r & 0 & 0 \\ -m^3\,r & m^2\,r^3 & \mathrm{Ra}\,m\,r^3 \\ 0 & 0 & -m^2\,r \end{pmatrix} y = 0,$$

qui est un système d'ÉDO linéaires du second ordre. On sait alors qu'il existe deux types de solutions au voisinage de 0 *: des solutions régulières et des solutions irrégulières. Comme l'étude de cette thèse porte sur le calcul des solutions régulières, nous allons alors déterminer les solutions régulières de ce système. Pour cela, écrivons le système ci-dessus avec la dérivation d'Euler* $\vartheta = x\frac{d}{dx}$*. Nous trouvons*

$$\begin{pmatrix} m & 0 & 0 \\ m & -r^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vartheta^2(y) + \begin{pmatrix} -2\,m & -r^2 & 0 \\ -2\,m & -r^2 & 0 \\ -1 & 0 & 0 \end{pmatrix} \vartheta(y) + \begin{pmatrix} -m^3\,r^2 & 0 & 0 \\ -m^3\,r^2 & m^2\,r^4 & \mathrm{Ra}\,m\,r^4 \\ 0 & 0 & -m^2\,r^2 \end{pmatrix} y = 0.$$

En calculant le déterminant de la matrice polynomiale $L_0(\lambda)$ *associée à ce dernier système, nous trouvons qu'il est non simple. En appliquant l'un des algorithmes élaborés au chapitre 3 pour calculer un système simple (ici nous utilisons l'algorithme* **EG_DV** *du chapitre 3), nous obtenons le système de première espèce*

$$\begin{pmatrix} m & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vartheta^2(y) + \begin{pmatrix} -2\,m & -r^2 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \vartheta(y) + \begin{pmatrix} -m^3\,r^2 & 0 & 0 \\ 0 & m^2\,r^2 & \mathrm{Ra}\,m\,r^2 \\ 0 & 0 & -m^2\,r^2 \end{pmatrix} y = 0.$$

Ce qui implique que la singularité $x = 0$ *est une singularité régulière et par suite le système (E.2) n'a que des solutions régulières. Finalement, en appliquant l'un des algorithmes du chapitre 2, nous obtenons une base de l'espace des solutions formelles de (E.2) donnée aux pages 63 et 64.*

■

Bibliography

[1] S. Abramov. EG - eliminations. *Journal of Difference Equations and Applications*, 5:393–433, 1999. 6, 10, 41, 65, 86, 160, 166

[2] S. Abramov and M. Bronstein. On solutions of linear functional systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 1–6. ACM Press, 2001. 6, 41, 65, 86, 160

[3] S. Abramov and M. Bronstein. Linear algebra for skew-polynomial matrices. Rapport de recherche 4420, INRIA, March 2002. 86

[4] S. Abramov, M. Bronstein, and D. E. Khmelnov. Regularization of linear recurrence systems. *Transactions of the A.M. Lyapunov Institute*, 4:158–171, 2003. 6, 10, 41, 68, 79, 86, 121, 160, 166

[5] S. Abramov, M. Bronstein, and D. E. Khmelnov. On regular and logarithmic solutions of ordinary linear differential systems. *Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, Springer Verlag*, 3718:1–12, 2005. 6, 7, 8, 41, 64, 160, 164

[6] A. Aparicio Monforte. *Méthodes Effectives pour l'Intégrabilité des Systèmes Dynamiques*. PhD thesis, Université de Limoges, December 2010. 159

[7] D. G. Babbitt and V. S. Varadarajan. Formal reduction theory of meromorphic differential equations: A group theoretic view. *Pacific Journal of Mathematics*, 109(1):1–80, 1983. 79, 111, 160

[8] W. Balser. *Formal Power Series and Linear Systems of Meromorphic Ordinary Differential Equations*. Springer, 2000. 5, 7, 36, 40, 72

[9] W. Balser, W. B. Jurkat, and D. A. Lutz. A general theory of invariants for meromorphic differential equations, Part I: Formal invariants. *Funcialaj Ekvacioj*, 22:197–221, 1979. 93, 160

[10] M. A. Barkatou. *Contribution à l'Étude des Équations Différentielles et aux Différences dans le Champ Complexe*. PhD thesis, Institut National Polytechnique de Grenoble, June 1989. 5, 37, 94

[11] M. A. Barkatou. An algorithm for computing a companion block diagonal form for a system of linear differential equations. *Applicable Algebra in Engineering, Communication and Computing*, 4:185–195, 1993. 5, 39, 94

[12] M. A. Barkatou. A rational version of Moser's algorithm. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 297–302. ACM Press, 1995. 5, 39, 97

[13] M. A. Barkatou. An algorithm to compute the exponential part of a formal fundamental matrix solution of a linear differential system. *Applicable Algebra in Engineering, Communication and Computing*, 8:1–23, 1997. 5, 94, 97, 111, 160

[14] M. A. Barkatou. On rational solutions of systems of linear differential equations. *Journal of Symbolic Computation*, 28:547–567, 1999. 5, 6, 9, 10, 11, 97, 121, 161, 164, 165, 167

[15] M. A. Barkatou. Factoring systems of linear functional equations using eigenrings. In I. S. Kotsireas and E. V. Zima, editors, *Latest Advances in Symbolic Algorithms*, pages 22–42. World Scientific, 2007. 5

[16] M. A. Barkatou, T. Cluzeau, and C. El Bacha. Algorithms for regular solutions of higher-order linear differential systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 7–14. ACM Press, 2009. 9, 38, 164

[17] M. A. Barkatou, T. Cluzeau, and C. El Bacha. Frobenius method for computing power series solutions of linear higher-order differential systems. In *Proceedings of the International Symposium on Mathematical Theory on Networks and Systems*, pages 1059–1066, Budapest, Hungary, 2010. 9, 38, 164

[18] M. A. Barkatou, T. Cluzeau, and C. El Bacha. Simple forms of higher-order linear differential systems and their applications in computing regular solutions. *Journal of Symbolic Computation*, 46(6):633–658, 2011. 9, 38, 68, 164

[19] M. A. Barkatou, C. El Bacha, G. Labahn, and E. Pflügel. On simultaneous row and column reduction of higher-order linear differential systems. *Journal of Symbolic Computation,* doi: 10.1016/j.jsc.2011.12.016, 2011. 13, 138, 169

[20] M. A. Barkatou, C. El Bacha, and E. Pflügel. Simultaneously row- and column-reduced higher-order linear differential systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 45–52. ACM Press, 2010. 12, 126, 168

[21] M. A. Barkatou and E. Pflügel. An algorithm computing the regular formal solutions of a system of linear differential equations. *Journal of Symbolic Computation*, 11:1–20, 1998. 5, 6, 7, 8, 9, 10, 11, 41, 64, 68, 69, 72, 73, 74, 75, 77, 95, 98, 160, 161, 164, 165, 167

[22] M. A. Barkatou and E. Pflügel. On the equivalence problem of linear differential systems and its application for factoring completely reducible systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 268–275. ACM Press, 1998. 5

[23] M. A. Barkatou and E. Pflügel. The ISOLDE package. A SourceForge Open Source project, `http://isolde.sourceforge.net`, 2006. 41, 64, 72

[24] M. A. Barkatou and E. Pflügel. Computing super-irreducible forms of systems of linear differential equations via Moser-reduction: A new approach. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 1–8. ACM Press, 2007. 10, 11, 77, 95, 165, 167

[25] M. A. Barkatou and E. Pflügel. On the Moser- and super-reduction algorithms of systems of linear differential equations and their complexity. *Journal of Symbolic Computation*, 44(8):1017–1036, 2009. 5, 10, 11, 77, 95, 98, 111, 157, 160, 165, 167

[26] M. A. Barkatou and F. Richard-Jung. Formal solutions of linear differential and difference equations. *Programming and Computer Software*, 23(1):17–30, 1997. 5, 94

[27] B. Beckermann, H. Chen, and G. Labahn. Fraction-free row reduction of matrices of Ore polynomials. *Journal of Symbolic Computation*, 41(5):513–543, 2006. 13, 86, 87, 138, 139, 140, 169

[28] B. Beckermann and G. Labahn. Recursiveness in matrix rational interpolation problems. *Journal of Computational and Applied Mathematics*, 77:5–34, 1997. 138, 139

[29] R. Ben Taher and M. Rachidi. Linear matrix differential equations of higher-order and applications. *Electronic Journal of Differential Equations*, 2008(95):1–12, 2008. 6, 160

[30] D. Boucher. *Sur les Équations Différentielles Linéaires Paramétrées, Une Application aux Systèmes Hamiltoniens*. PhD thesis, Université de Limoges, October 2000. 7, 159

[31] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Grundlehren der Mathematischen Wissenschaften 315, Springer-Verlag, Berlin, 1997. 89

[32] R. Byers, V. Mehrmann, and H. Xu. Trimmed linearizations for structured matrix polynomials. *Linear Algebra and its Applications*, 429:2373–2400, 2008. 22

[33] G. Chen. *Solutions Formelles de Systèmes d'Équations Différentielles Linéaires Ordinaires Homogènes*. PhD thesis, Université Joseph Fourrier - Grenoble I, February 1990. 94

[34] B. D. Choi, Y. C. Kim, and Y. W. Lee. The M/M/c retrial queue with geometric loss and feedback. *Computers Math. Applic.*, 36(6):41–52, 1998. 5, 159

[35] R. C. Churchill and J. J. Kovacic. Cyclic vectors. In *Differential Algebra and Related Topics, Proceedings of the International Workshop*, pages 191–218, Rutgers University, Newark, November 2002. River Edge, NJ, World Scientific Publishing Company. 39, 94

[36] T. Cluzeau. *Algorithmique Modulaire des Équations Différentielles Linéaires*. PhD thesis, Université de Limoges, September 2004. 5, 39, 94

[37] E. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill Book Company, Inc., New York, 1955. 5, 7, 36, 37, 38, 39, 40, 42, 57, 72, 162

[38] P. M. Cohn. *Free Rings and their Relations*. Academic Press, London & New York, 1971. 6, 86, 139, 161

[39] F. Cope. Formal solutions of irregular linear differential equations, Part II. *Amer. J. Math.*, 58:130–140, 1936. 39

[40] P. Davies, H. Cheng, and G. Labahn. Computing Popov form of general Ore polynomial matrices. In *Milestones in Computer Algebra*, pages 149–156, 2008. 6, 137, 151, 161

[41] F. De Terán and F. M. Dopico. Sharp lower bounds for the dimension of linearizations of matrix polynomials. *Electronic Journal of Linear Algebra*, 17:518–531, 2008. 22

[42] F. De Terán, F. M. Dopico, and D. S. Mackey. Linearizations of singular matrix polynomials and the recovery of minimal indices. *Electronic Journal of Linear Algebra*, 18:371–402, 2009. 7, 10, 32, 161, 165

[43] A. Fredet. *Résolution sous Forme Finie d'Équations Différentielles Linéaires et Extensions Exponentielles*. PhD thesis, École Polytechnique, November 2001. 159

[44] G. Frobenius. Über die integration der linearen differentialgleichungen mit veränder koefficienten. *Journal für die reine und angewandte Mathematik*, 76:214–235, 1873. 7, 8, 37, 57, 159, 162, 164

[45] F. R. Gantmacher. *Théorie des Matrices, I & II*. Dunop, Paris, 1966. 17, 22, 32

[46] C. W. Gear. Differential-algebraic equation index transformation. *SIAM J. Sci. Stat. Comput.*, 9(1):39–47, 1988. 6, 125, 161

[47] C. W. Gear and L. R. Petzold. ODE methods for the solution of differential/algebraic systems. *SIAM J. Numer. Anal.*, 21(4):716–728, 1984. 12, 168

[48] K. O. Geddes, S. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Publishing, Boston, 1992. 150

[49] M. Giesbrecht and M. S. Kim. On computing the Hermite form of a matrix of differential polynomials. In *Computer Algebra and Scientific Computation (CASC) Workshop*, Lecture Notes in Computer Science 5743, pages 118–129, 2009. 6, 137, 161

[50] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982. 7, 8, 17, 19, 20, 21, 23, 25, 71, 161, 164

[51] G.-M. Greuel, V. Levandovskyy, and H. Schönemann. PLURAL. A SINGULAR 3.0 subsystem for computations with non-commutative polynomial algebras. Centre for Computer Algebra, University of Kaiserslautern, `http://www.singular.uni-kl.de`, 2006. 6, 161

[52] W. A. Harris, Y. Sibuya, and L. Weinberg. A reduction algorithm for linear differential systems. *Funkcialaj Ekvacioj*, 11:59–67, 1968. 13, 126, 127, 128, 129, 130, 168, 169

[53] L. Heffter. *Einleitung in die Theorie der Linearen Differentialgleichungen*. Teubner, Leipzig, 1894. 7, 41, 162

[54] A. Hilali. *Contribution à l'Étude des Points Singuliers des Systèmes Différentiels Linéaires*. PhD thesis, L'université scientifique et médicale de Grenoble, Avril 1982. 5, 8, 39, 40, 57, 164

[55] A. Hilali. Calcul des invariants de Malgrange et de Gérard-Levelt d'un système différentiel linéaire en un point singulier irrégulier. *Journal of Differential Equations*, 69:404–421, 1987. 5, 97

[56] A. Hilali. *Solutions Formelles de Systèmes Différentiels Linéaires au Voisinage d'un Point Singulier*. PhD thesis, L'université scientifique, technologique et médicale de Grenoble, June 1987. 5, 7, 8, 39, 72, 97, 160, 164

[57] A. Hilali and A. Wazner. Un algorithme de calcul de l'invariant de Katz d'un système différentiel linéaire. *Annales de l'institut Fourier*, 36(3):67–81, 1986. 5

[58] A. Hilali and A. Wazner. Formes super-irréductibles des systèmes différentiels linéaires. *Numerische Mathematik*, 50(4):429–449, 1987. 10, 11, 12, 77, 95, 97, 98, 103, 111, 121, 157, 160, 165, 167, 168

[59] E. L. Ince. *Ordinary Differential Equations*. Dover Publications, Inc., 1926. 5, 37, 57

[60] A. Jameson. Solution of the equation $AX + XB = C$ by inversion of an $M \times M$ or $N \times N$ matrix. *SIAM J. Appl. Math.*, 16(5):1020–1023, 1968. 40

[61] C.-P. Jeannerod and G. Villard. Straight-line computation of the polynomial matrix inverse. Research Report 2002-47, Laboratoire LIP, ENS Lyon, France, 2002. 53

[62] C.-P. Jeannerod and G. Villard. Asymptotically fast polynomial matrix algorithms for multivariable systems. *International Journal of Control*, 79(11):1359–1367, 2006. 19, 33

[63] L. Jódar, R. Company, and E. Navarro. Laguerre matrix polynomials and systems of second-order differential equations. *Applied Numerical Mathematics*, 15:53–63, 1994. 41

[64] L. Jódar and M. Legua. Solving second-order matrix differential equations with regular singular points avoiding the increase of the problem dimension. *Applied Mathematics and Computation*, 53:191–205, 1993. 6, 7, 8, 41, 57, 160, 164

[65] T. Kailath. *Linear Systems*. Prentice-Hall, 1980. 7, 10, 17, 32, 138, 139, 161, 165

[66] G. I. Kalogeropoulos, A. D. Karageorgos, and A. A. Pantelous. Higher-order linear matrix descriptor differential equations of Apostol-Kolodner type. *Electronic Journal of Differential Equations*, 2009(25):1–13, 2009. 6, 160

[67] P. Kunkel and V. Mehrmann. Canonical forms for linear differential-algebraic equations with variable coefficients. *Journal of Computational and Applied Mathematics*, 56:225–251, 1994. 6, 12, 125, 126, 161, 168

[68] P. Lancaster. Linearization of regular matrix polynomials. *Electronic Journal of Linear Algebra*, 17:21–27, 2008. 21

[69] P. Lancaster and M. Tismenetsky. *The Theory of Matrices, Second Edition: With Applications*. Academic Press, London, 1985. 17, 22

[70] V. Levandovskyy and K. Schindelar. Computing diagonal form and Jacobson normal form of a matrix using Gröbner bases. *Journal of Symbolic Computation*, 46(5):595–608, 2011. 6, 138, 161

[71] D. L. Littlefield and P. V. Desai. Frobenius analysis of higher order equations: Incipient buoyant thermal convection. *SIAM J. Appl. Math.*, 50(6):1752–1763, 1990. 5, 62, 159

[72] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. of Matrix Analysis and Appl.*, 28(4):971–1004, 2005. 21

[73] V. Mehrmann and C. Shi. Transformation of higher order differential-algebraic systems to first order. *Numerical Algorithms*, 42:281–307, 2006. 5, 6, 137, 159, 161

[74] J. Middeke. A polynomial-time algorithm for the Jacobson form for matrices of differential operators. Technical Report 08-13, Research Institute for Symbolic Computation (RISC) Report Series, 2008. 6, 138, 161

[75] M. Miyake. Remarks on the formulation of the Cauchy problem for general system of ordinary differential equations. *Tôhoku Mathematical Journal*, 32:79–89, 1980. 87

[76] T. Mora. *Solving Polynomial Equation Systems I : The Kronecker-Duval Philosophy*. Encyclopedia of Mathematics and its applications 88, Cambridge University Press, 2003. 19

[77] J. Moser. The order of a singularity in Fuchs' theory. *Mathematische Zeitschrift*, 72:397–398, 1960. 5, 12, 39, 79, 97, 109, 121, 136, 157, 168

[78] T. Mulders and A. Storjohann. On lattice reduction for polynomial matrices. *Journal of Symbolic Computation*, 35:377–401, 2003. 14, 138, 170

[79] E. Navarro, R. Company, and L. Jódar. Bessel matrix differential equations: Explicit solutions of initial and two-point boundary value problems. *Applicationes Mathematicae*, 22(1):11–23, 1993. 6, 7, 41, 160

[80] E. Navarro, R. Company, and L. Jódar. Solving higher order Fuchs type differential systems avoiding the increase of the problem dimension. *International Journal of Mathematics and Mathematics Sciences*, 17(1):91–102, 1994. 6, 7, 8, 41, 57, 160, 164

[81] E. Navarro, M. V. Ferrer, and L. Jódar. A matrix method of Frobenius for solving implicit second order differential systems. *Analysis*, 13:259–278, 1993. 5, 7, 8, 57, 159, 164

[82] A. Pantelous, A. Karageorgos, and G. Kalogeropoulos. Power series solutions for linear higher order rectangular differential matrix control systems. In *17th Mediterranean Conference on Control and Automation*, pages 330–335, Makedonia Palace, Thessaloniki, Greece, 2009. 5, 137, 159

[83] E. Pflügel. An algorithm for computing exponential solutions of first order linear differential systems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 164–171. ACM Press, 1997. 5

[84] E. Pflügel. *Résolution Symbolique des Systèmes Différentiels Linéaires*. PhD thesis, Université Joseph Fourrier, October 1998. 94, 95

[85] E. Pflügel. Effective formal reduction of linear differential systems. *Applicable Algebra in Engineering, Communication and Computing*, 10:153–187, 2000. 5, 6, 10, 11, 94, 95, 96, 98, 160, 161, 166, 167

[86] E. G. C. Poole. *Introduction to the Theory of Linear Differential Equations*. Oxford University Press, London, 1936. 7, 8, 37, 38, 49, 162

[87] M.-P. Quéré-Stuchlik. *Algorithmique des Faisceaux Linéaires de Matrices, Applications à la Théorie des Systèmes Linéaires et à la Résolution d'Équations Algébro-Différentielles*. PhD thesis, LMC-IMAG, 1996. 126

[88] M.-P. Quéré-Stuchlik and G. Villard. An algorithm for the reduction of linear DAE. In *Proceedings of the International Symposium on Symbolic and Algebraic Computations*, pages 223–231. ACM Press, 1995. 6, 12, 126, 161, 168

[89] S. Schulz. Four lectures on differential algebraic equations. Technical Report 497, The University of Auckland, 2003. 5, 6, 125, 137, 159, 161

[90] C. Shi. *Linear Differential-Algebraic Equations of Higher-Order and the Regularity or Singularity of Matrix Polynomials*. PhD thesis, von der Fakultät II - Mathematik und Naturwissenschaften der Technischen Universität Berlin, July 2004. 6, 78, 161

[91] M. Singer. Testing reducibility of linear differential operators: A group theoretic perspective. *Applicable Algebra in Engineering, Communication and Computing*, 7(2):77–104, 1996. 159

[92] L. Sirovich. *Techniques of Asymptotic Analysis*. Springer-Verlag New York, Heidelberg, Berlin, 1971. 8, 57, 164

[93] A. Storjohann and G. Villard. Computing the rank and a small nullspace basis of a polynomial matrix. Research Report, LIP, ENS Lyon, France, 2005. 19

[94] E. Tournier. *Solutions Formelles d'Équations Différentielles, le Logiciel de Calcul Formel: DÉSIR, Étude Théorique et Réalisation*. PhD thesis, L'université scientifique, technologique et médicale de Grenoble, April 1987. 5, 8, 57, 94, 159, 164

[95] H. L. Turritin. Convergent solutions of ordinary linear homogeneous differential equations in the neighborhood of an irregular singular point. *Acta Mathematica*, 93:27–66, 1955. 93

[96] M. Van Barel and A. Bultheel. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numerical Algorithms*, 3:451–462, 1992. 33

[97] P. M. Van Dooren, P. Dewilde, and J. Vandewalle. On the determination of the Smith-MacMillan form of a rational matrix from its Laurent expansion. *IEEE Trans. Circuit Systems*, 26:180–189, 1979. 17, 26

[98] M. Van Hoeij. Factorization of differential operators with rational functions coefficients. *Journal of Symbolic Computation*, 24:537–561, 1997. 159

[99] M. Van Hoeij. Formal solutions and factorization of differential operators with power series coefficients. *Journal of Symbolic Computation*, 24:1–30, 1997. 159

[100] J. Von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, 2003. 1, 19

[101] W. Wasow. *Asymptotic Expansions for Ordinary Differential Equations*. Dover Publications Inc., New York, 1987. 5, 7, 36, 40, 93, 94

[102] J. Wilkening. An algorithm for computing Jordan chains and inverting analytic matrix functions. *Linear Algebra Appl.*, 427(1):6–25, 2007. 29

[103] W. Zhou and G. Labahn. Efficient order basis computation. In *Proceedings of the International Symposium on Symbolic and Algebraic Computations*, pages 375–382. ACM Press, 2009. 33

[104] J. C. Zúniga. *Algorithmes Numériques pour les Matrices Polynomiales avec Applications en Commande*. PhD thesis, Institut National des Sciences Appliquées de Toulouse, September 2005. 7, 17, 26, 27, 29, 32, 161

Résumé : Dans cette thèse, nous développons de nouvelles méthodes algébriques pour la résolution d'une classe importante de systèmes d'équations différentielles linéaires d'ordre arbitraire. De tels systèmes apparaissent dans de nombreuses disciplines scientifiques comme la chimie, la physique, la mécanique et la théorie du contrôle.

Dans un premier temps, nous nous intéressons à l'analyse locale des systèmes d'équations différentielles linéaires ordinaires au voisinage d'une singularité. Nous développons des algorithmes pour le calcul des solutions régulières formelles. Ces algorithmes sont directs, c'est-à-dire, ne transforment pas le système en un autre du premier ordre et de taille plus grande. Nos approches sont fondées sur l'utilisation des propriétés des matrices polynomiales dont le déterminant joue le même rôle que les polynômes indiciels dans le cas scalaire. Puis, nous nous intéressons à l'étude des formes dites k-simples d'un système différentiel linéaire explicite du premier ordre. Ces formes donnent des informations sur les pentes entières du polygone de Newton du système et permettent de calculer les solutions formelles sans ramification. Notre contribution se reflète par le développement d'une méthode directe pour le calcul des formes k-simples.

Dans un second temps, nous étudions les systèmes d'équations algébro-différentielles linéaires. De tels systèmes sont composés d'équations différentielles ordinaires couplées à des équations purement algébriques. Nous proposons des algorithmes pour les découpler en une partie purement différentielle et une autre purement algébrique.

Une autre contribution de la thèse est l'étude de la complexité et l'implémentation en Maple de nos divers algorithmes mis en œuvre.

Mots-clés : Équations différentielles matricielles, singularités, solutions formelles, calcul algébrique, algorithmes de réduction.

Abstract: In this thesis, we develop new algebraic methods for solving an important class of systems of linear differential equations of arbitrary order. Such systems appear in many scientific disciplines such as chemistry, physics, mechanics and control theory.

Firstly, we are interested in the local analysis of systems of linear ordinary differential equations near a singularity. We develop algorithms for the computation of their regular formal solutions. These algorithms are direct, *i.e.*, do not reduce the system to another one of first-order and larger size. Our approaches use properties of matrix polynomials whose determinant plays the same role as indicial polynomials in the scalar case. Then, we are interested in the study of the so-called k-simple forms of a linear differential system of first-order. These forms give information on the integer slopes of the Newton polygon of the system and allow the computation of the formal solutions without ramification. Our contribution is reflected in the development of a direct method for the calculation of k-simple forms.

Secondly, we focus on systems of linear differential-algebraic equations. Such systems are composed of differential equations coupled with algebraic ones. We propose algorithms for decoupling them into a purely differential part and a purely algebraic one.

Another contribution of the thesis is the study of the complexity and the implementation in Maple of the algorithms developed.

Keywords: Matrix differential equations, singularities, formal solutions, computer algebra, reduction algorithms.